

TESIS DOCTORAL

The logo of the Universidad Nacional de Educación a Distancia (UNED), consisting of the letters 'UNED' in white on a dark green square background.

AÑO 2014

**“IMPLEMENTACIÓN DE REGLAS DE
PROYECCIÓN CONCEPTUALES EN EL MARCO DE LA
SEMÁNTICA PROFUNDA PARA LA REUTILIZACIÓN
DE BASES DE CONOCIMIENTO ENCICLOPÉDICO”**

Autora: María de los Llanos Carrión Varela

Licenciada en Filología Inglesa

Máster en Lingüística Inglesa Aplicada

**UNIVERSIDAD NACIONAL DE EDUCACIÓN A
DISTANCIA**

Departamento de Filologías Extranjeras y sus Lingüísticas

Facultad de Filología

Director: Dr. D. Ricardo Mairal Usón

Codirector: Dr. D. J. Carlos Perriñán Pascual

Departamento de Filologías Extranjeras y sus Lingüísticas

Facultad de Filología

**“IMPLEMENTACIÓN DE REGLAS DE
PROYECCIÓN CONCEPTUALES EN EL MARCO DE LA
SEMÁNTICA PROFUNDA PARA LA REUTILIZACIÓN
DE BASES DE CONOCIMIENTO ENCICLOPÉDICO”**

Autora: María de los Llanos Carrión Varela

Licenciada en Filología Inglesa

Máster en Lingüística Inglesa Aplicada

Director: Dr. D. Ricardo Mairal Usón

Codirector: Dr. D. J. Carlos Periñán Pascual

AGRADECIMIENTOS

En primer lugar, quisiera mostrar mi agradecimiento tanto a mi director de tesis D. Ricardo Mairal Usón, por su valiosa guía y su apoyo durante estos años, como a mi co-director D. J. Carlos Perrián Pascual, por su inestimable ayuda y por la minuciosidad en sus observaciones. Igualmente, quisiera agradecerles a ambos el haber confiado en mí para la realización de este trabajo y los ánimos que siempre me han insuflado en los momentos clave de su elaboración, sin los cuales no habría podido ser capaz de completarlo.

También quisiera mostrar mi gratitud a los profesores de la Universidad de Murcia que desde un principio me inspiraron la pasión por la lingüística, D. José Miguel Hernández Terrés y D. Javier Valenzuela, y a la profesora de lengua española que en el I.E.S. Bachiller Sabuco de Albacete me descubrió el apasionante mundo de las letras, Dña. María Antonia Sanabria. Todos ellos me enseñaron mucha materia académica pero, lo que es más importante, fueron especialmente capaces de transmitirme una pasión por el lenguaje que me ha acompañado sin abandonarme a lo largo de toda mi vida.

A todos los miembros de los grupos de investigación Lexicom y SECONDS y a todos los colegas de FunGramKB, muchas gracias por la calurosa acogida y el amable trato que me habéis brindado siempre, tanto en lo académico como en lo personal. Es un placer poder colaborar con profesionales de vuestra talla, de los que tanto he aprendido.

A mi familia, mis padres, mis hermanos y sus consortes, a quienes más que gracias debería decir también lo siento: por tantos ratos de quejas y zozobras que habéis soportado y por tantos momentos en los que no he estado presente; gracias por

apoyarme siempre cuando lo he necesitado. En especial, merecen mi agradecimiento más profundo e incondicional mis padres, sin los cuales no sería ni una sombra de lo que soy hoy.

A toda mi familia política, en especial a Esperanza, por todo lo que me has enseñado sobre la lengua y por estar siempre ahí para resolver dudas, ya fueran lingüísticas o existenciales.

A mis amigos y muy en especial a mis amigas, quienes también se han cargado a hombros su buena parte de esta tesis: gracias por empeñaros en sacarme de mi cueva cuando parecía que la oscuridad iba a engullirme por completo.

A todos los investigadores sobre los que se sustenta mi trabajo.

Finalmente, pero no por ello menos importante, quisiera mostrar mi agradecimiento a Álvaro, por no tirar nunca la toalla y por apoyarme siempre desde el principio y en todo momento. En un mundo de variables, tú has sido y eres mi constante.

Y a todos los que seguramente olvido sin que lo merezcan, a los que desde este momento pido disculpas,

MUCHAS GRACIAS.

ÍNDICE

CAPÍTULO I	INTRODUCCIÓN	13
1.1.	INTRODUCCIÓN	13
1.2.	LA MIGRACIÓN DE DATOS.....	15
1.3.	OBJETIVOS Y ESTRUCTURA DE LA PRESENTE TESIS DOCTORAL 19	
CAPÍTULO II	ONTOLOGÍAS Y BASES DE CONOCIMIENTO	23
2.1.	INTRODUCCIÓN.....	23
2.2.	ONTOLOGÍA VS. BASE DE CONOCIMIENTO: DEFINICIONES.....	25
2.3.	LA MIGRACIÓN DE DATOS EN LAS ONTOLOGÍAS Y BASES DE CONOCIMIENTO	31
2.4.	FUNDAMENTOS LINGÜÍSTICOS DE LAS ONTOLOGÍAS: SEMÁNTICA PROFUNDA VS. ENFOQUE RELACIONAL.....	40
CAPÍTULO III	FUNGRAMKB Y EL ONOMASTICÓN	47
3.1.	¿QUÉ ES FUNGRAMKB?.....	47
3.2.	COREL COMO LENGUAJE DE REPRESENTACIÓN: LOS POSTULADOS DE SIGNIFICADO Y LOS MARCOS TEMÁTICOS	57
3.3.	EL ONOMASTICÓN DE FUNGRAMKB: ARQUITECTURA Y CONTENIDO	63
3.4.	ESTRUCTURAS RELACIONADAS: YAGO, YAGO2 Y DBPEDIA	69
3.4.1.	YAGO.....	71
3.4.2.	YAGO2.....	73
3.4.3.	DBPEDIA.....	81
CAPÍTULO IV	METODOLOGÍA DE POBLACIÓN DEL ONOMASTICÓN ...	92
4.1.	PLANTEAMIENTO PARA LA METODOLOGÍA	92
4.2.	LAS PROPIEDADES EN DBPEDIA: LA JERARQUÍA DE CLASES	93

4.3. LAS FASES DEL PROCESO DE CREACIÓN DE REGLAS	97
4.3.1. LA FASE DE CONSTRUCCIÓN	98
4.3.1.1. Relevancia y pertinencia de la propiedad para FunGramKB	98
4.3.1.2. Identificación del tipo de bio-estructura (i.e. retrato o historia) a la que pertenecerá la predicación	101
4.3.1.3. En caso de valor numérico, identificación de la unidad de medida en la que se expresa el valor de la propiedad	106
4.3.1.4. Construcción de las predicaciones de la regla en lenguaje COREL	114
4.3.2. LA FASE DE PROYECCIÓN	125
4.3.3. LA FASE DE ACTUALIZACIÓN	128
CAPÍTULO V ANÁLISIS DE RESULTADOS	130
5.1. PRESENTACIÓN	130
5.2. REGLAS ELABORADAS PARA EL PRESENTE TRABAJO	133
5.3. ANÁLISIS DE LA CLASIFICACIÓN ORIGINAL	135
5.4. ALTERNATIVAS TAXONÓMICAS PARA CLASIFICAR LAS REGLAS ELABORADAS PARA EL ONOMASTICÓN DE FUNGRAMKB	141
5.4.1. LA CLASIFICACIÓN FORMAL DE LAS REGLAS DEL ONOMASTICÓN: TIPOS DE ESTRUCTURAS	143
5.4.1.1. La estructura CONTEXT	144
5.4.1.2. La estructura ATTRIBUTE	145
5.4.1.3. La estructura EV+SAT	146
5.4.1.4. La estructura EVENT ROLE	150
5.4.1.5. Las estructuras TANDEM-CROSS REF, EMBED CONDITION y RANGE-IN-SAT: las preferencias de selección	151
5.4.1.6. La estructura GENITIVE	159
CAPÍTULO VI APLICACIONES DEL ONOMASTICÓN PARA RESOLVER PROBLEMAS LINGÜÍSTICOS	165
6.1. LA AMBIGÜEDAD EN EL LENGUAJE	165
6.2. LA REFERENCIA Y CORREFERENCIA	166

6.3. LA ANÁFORA	172
6.3.1. LA RESOLUCIÓN DE LA ANÁFORA INDIRECTA	176
6.4. LA METÁFORA Y LA METONIMIA.....	180
6.4.1. LA <i>METAFTONIMIA</i>	188
CAPÍTULO VII CONCLUSIONES	197
7.1. RESUMEN DE LA PRESENTE TESIS DOCTORAL	197
7.2. CUESTIONES TRANSVERSALES DE LA MIGRACIÓN DE DATOS	201
7.3. CONCLUSIONES, APLICACIONES Y FUTURA INVESTIGACIÓN	203
BIBLIOGRAFÍA	208
APÉNDICE I	
Listado de reglas y propiedades en el Onomasticón de FunGramKB	222
APÉNDICE II 242	
Estadísticas totales de la población del Onomasticón.....	242

SÍMBOLOS, ABREVIATURAS Y SIGLAS

a/n: *alternative name*

ATRIB SIMPLES: Reglas de tipo semántico *atributivas simples*

ATRIB ABSOL: Reglas de tipo semántico *atributivas absolutas*

COREL: *Conceptual Representation Language*

HTML: *HyperText Markup Language*

MLC: Modelo Léxico Construccional

n/r: *no rule*

OWL: *Ontology Web Language*

PLN: Procesamiento del Lenguaje Natural

PRED REL: Reglas de tipo semántico *predicativas relacionales*

PRED INT: Reglas de tipo semántico *predicativas intrínsecas*

RDF: *Resource Description Framework*

SWD: *Semantic Web Document*

W3C: *World Wide Web Consortium*

WWW: *World Wide Web*

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de FunGramKB	49
Figura 2. Ejemplo de estructuración ontológica en FunGramKB (Periñán Pascual y Arcas Túnez, 2007a:199).	54
Figura 3. El procedimiento de la metodología COHERENT (Periñán Pascual y Mairal Usón, 2011:20).	55
Figura 4. El Planeta Cognitivo (Periñán Pascual y Mairal Usón, 2010b).	65
Figura 5. Ilustración de la línea temporal generada para Abraham Lincoln mostrando su conexión con otras personas identificadas (Bhole <i>et al.</i> , 2007:467).	80
Figura 6. <i>Linked Open Data</i> en la web (Jentzsch, 2009).	87
Figura 7. Arquitectura básica del sistema PHYTHIA (Unger y Cimiano, 2011:154). ...	88
Figura 8. Fragmento de la Ontología de DBpedia (Lehmann <i>et al.</i> , en prensa).	93
Figura 9. Interfaz del DBpedia Mapper en el editor del Onomasticón.	106
Figura 10. Proyección de la instanciación “The Da Vinci Code” en la regla para la propiedad <i>publisher</i>	127
Figura 11. Conexión conceptual para la resolución metonímico-metafórica de la expresión “Ford”	187
Figura 12. Intersección de los dominios como meta para la metonimia ENTIDAD POR ZONA ACTIVA (Brdar y Brdar-Szabó, 2007:132).	191

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Vista parcial de FunGramKB Suite, mostrando los diferentes módulos y algunos recursos lexicográficos disponibles	49
Ilustración 2. Ficha informativa (<i>Info-box</i>) de Wikipedia (Periñán Pascual y Carrión Varela, 2011:93).	82
Ilustración 3. La clase “Place” y sus diversas sub-clases y propiedades en el Onomasticón de FunGramKB.	94
Ilustración 4. Propiedad <i>elevation</i> , dentro de la clase “Place”	95
Ilustración 5. Propiedad <i>areaTotal</i> , dentro de la clase “Place”	96
Ilustración 6. Propiedad <i>nearestCity</i> , dentro de la clase “Place”	96
Ilustración 7. Propiedad <i>elevation</i> dentro de la clase “Place”	102
Ilustración 8. Propiedad <i>child</i> dentro de la clase “Person”	103
Ilustración 9. Propiedad <i>children</i> en Schema.org	104
Ilustración 10. Búsqueda de la unidad léxica “metro” en FunGramKB Editor del Onomasticón.	107
Ilustración 11. Búsqueda de la unidad léxica “metro” en <i>FunGramKB Editor</i> del Onomasticón.	108

Ilustración 12. Jerarquía del concepto básico +UNIT_OF_SPACE_00.	110
Ilustración 13. Propiedad <i>address</i> dentro de la clase “Building”	112
Ilustración 14. Búsqueda infructuosa de la unidad léxica “elevation” en <i>FunGramKB Editor</i> del Onomasticón.	116
Ilustración 15. Identificación del concepto \$ALTITUDE_00 en FunGramKB a través de la lexicalización “height”.	117
Ilustración 16. La propiedad <i>publisher</i> en DBpedia	118
Ilustración 17. La propiedad <i>publisher</i> en Schema.org	118
Ilustración 18. El concepto \$PUBLISH_00 en la Ontología de FunGramKB.	119
Ilustración 19. Jerarquía conceptual bajo el concepto +GROUP_00.	123
Ilustración 20. Jerarquía conceptual en la que se encuadra el nuevo concepto \$PARTY_01.	123
Ilustración 21. Introducción de la regla para la propiedad <i>elevation</i> en FunGramKB Editor del Onomasticón.....	125

ÍNDICE DE TABLAS

Tabla 1. Tipología de esquemas conceptuales en FunGramKB (Periñán Pascual y Arcas Túnez, 2010b:2668).	65
Tabla 2. Lista de propiedades de “City” con ausencia de productividad semántica y conceptual.....	99
Tabla 3. El nuevo concepto \$SQUARE_METRE_00.	111
Tabla 4. La propiedad <i>address</i> de la clase “Building” en el Onomasticón de FunGramKB.	112
Tabla 5. La propiedad <i>nearestCity</i> de la clase “Place” en el Onomasticón de FunGramKB.	113
Tabla 6. La propiedad <i>publisher</i> de la clase “Work”	120
Tabla 7. Número de entidades dentro de las clases más pobladas en DBpedia	134
Tabla 8. Tipos de propiedades atendiendo a criterios de clasificación semánticos.	141
Tabla 9. La propiedad <i>birthDate</i> de la clase “Person”	146
Tabla 10. La propiedad <i>subsequentWork</i> de la clase “Work”	148
Tabla 11. La propiedad <i>day</i> de la clase “City”	149
Tabla 12. La propiedad <i>residence</i> de la clase “Person”	150
Tabla 13. La propiedad <i>locationCity</i> de la clase “Organisation”.	153
Tabla 14. La propiedad <i>keyPersonPosition</i> de la clase “Organisation”.	153
Tabla 15. La propiedad <i>language</i> de la clase “Populated Place”.	156
Tabla 16. La propiedad <i>architecturalStyle</i> de la clase “Building”.	157
Tabla 17. Las propiedades <i>location</i> , <i>locationCity</i> y <i>locationCountry</i> de la clase “Organisation”.	158
Tabla 18. La propiedad <i>father</i> de la clase “Person”	160
Tabla 19. Tipos de estructuras atendiendo a criterios de clasificación formales.	162
Tabla 20. Frecuencia de aparición de cada estructura formal en los diferentes tipos semánticos de propiedades.	164

Tabla 21. El concepto +FATHER_00 en la Ontología de FunGramKB.	174
Tabla 22. El subconcepto -DRINK en la Ontología de FunGramKB.....	184

*“The Web as I envisaged it, we have not seen it yet.
The future is still so much bigger than the past.”*

“You affect the world by what you browse.”

Tim Berners-Lee, creador de la World Wide Web

CAPÍTULO I

INTRODUCCIÓN

1.1. INTRODUCCIÓN

La búsqueda de respuestas y el descubrimiento de la verdad han sido eternos ideales para el ser humano desde el principio de los tiempos. Con este objetivo por bandera, el mundo de la ciencia ha dedicado ímprobos esfuerzos para aumentar no sólo su saber, sino la capacidad de averiguar lo que a nuestros ojos puede resultar oculto o desconocido. Siguiendo esta estela a través de los siglos, y cada vez más en nuestro tiempo y sociedad actuales, la portentosa capacidad, antaño velada, que ahora tiene el ser humano de obtener información se perfila como el gran bien de nuestra era. El mundo alfabetizado presume de ser el más competente de todos los tiempos, debido al masivo flujo de conocimiento y datos que, gracias a herramientas como Internet, circulan globalmente, dando lugar a la ubicua “sociedad de la información”, como se ha denominado al mundo digital de nuestro tiempo. Sin embargo, a pesar de manejar esta ingente profusión de datos, las máquinas que los procesan no son capaces de comprenderlos. Se limitan a transfundirlos entre usuarios, a proporcionarlos bajo

demanda humana, sin tener la capacidad de filtrar o depurar aquellos que en realidad son relevantes para el interesado que formula la consulta, con la consiguiente pérdida de tiempo que ello acarrea. El propio inventor de la World Wide Web (WWW), Tim Berners-Lee, junto a otros dos autores, James Hendler y Ora Lassila, ya reconocían en su momento que *“Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully.”* (Berners-Lee *et al.*, 2001:3), lo que cobra más sentido si recordamos las palabras que el creador de la WWW ya enunciaba en los albores del lanzamiento global de dicha herramienta, cuando se refería a las posibilidades de investigación que se abrían tras la implementación de la misma: *“Among these are new human interface techniques for managing a large space and the user's view of it, and automatic tools for traversing the web and searching indexes in pursuit of the answers to specific questions.”* (Berners-Lee *et al.*, 1992:459). De este modo se presentaba la principal inquietud que los usuarios de la red, al igual que el ser humano desde los albores de su nacimiento, albergarían sin descanso durante los años venideros y hasta nuestros días: la incesante búsqueda de respuestas a interrogantes específicos.

A la hora de encaminarse hacia el descubrimiento de dichas respuestas, cuando los usuarios se valen de los actuales motores de búsqueda más populares en la red (por ejemplo, Google¹), constatan, no sin desánimo, que éstos carecen de lo que se denomina conocimiento semántico: son capaces de identificar y reconocer una cadena de símbolos gráficos, letras, que forman palabras, mas sin poseer la capacidad de discernir qué se esconde tras esos símbolos o qué persigue el usuario mediante su búsqueda. ¿Quién no ha experimentado, como consumidor de un motor de búsqueda, un tedioso sentimiento de frustración al no ser capaz de hacer que la máquina entienda qué es lo que estamos

¹ <http://www.google.com>

buscando? Obviamente, el dotar de contenido semántico a la información presente en la red, que para la máquina son meros símbolos despojados de significado, supone adentrarse en los dominios de la Inteligencia Artificial. No obstante, los usos inmediatos de los que la sociedad en general lograría beneficiarse podrían resumirse en palabras de Giles (2008:26): *“Users of the semantic web will be able to ask questions rather than search for phrases”*.

1.2. LA MIGRACIÓN DE DATOS

Para acometer la tarea mencionada anteriormente, es decir, lograr que las máquinas sean capaces de aportar soluciones concretas a consultas o problemas determinados, es necesario que dichas máquinas sean capaces, en primer lugar, de intercambiar información entre ellas de la manera adecuada. Esto significa que, para que las computadoras tengan la capacidad de intercambiar datos, éstos deben ser compatibles entre sí y adecuados a las pasarelas que permiten su transferencia. Este traspaso de contenidos entre diferentes sistemas y herramientas es lo que denominamos *migración de datos*.

La migración de datos constituye una práctica que maximiza la eficiencia y utilidad de las herramientas de PLN, ya que permite optimizar tanto el tiempo como el esfuerzo invertido previamente en elaborar o recopilar una fuente de datos determinada. Asimismo, la reutilización de los recursos permite en numerosos casos que las fuentes de datos que han servido de origen se vean enriquecidas y mejoradas. Esto sucede cuando el recurso de destino de los contenidos efectúa un tratamiento de los mismos que mejora su organización o, incluso, que facilita la posterior reutilización de ellos por parte de terceros. Por tanto, encontramos que lo más importante de los procesos de migración de datos no estriba en el cambio de formato de éstos, sino en el producto final

resultante, el cual puede llegar a ser mucho más rico conceptualmente que el repositorio de origen.

En numerosas ocasiones, el origen de la práctica de la migración de datos está arraigado en cuestiones logísticas de las organizaciones. Por ejemplo, es muy habitual encontrar que una empresa desea modernizar sus sistemas informáticos pero que no está dispuesta a perder toda la información antigua que está almacenada en sus archivos. En este caso, hablamos de la migración de datos de *sistemas heredados* a nuevos sistemas más modernos. Así pues, tomando como referencia este traspaso de contenidos entre sistemas, podemos definir la migración de datos como indican Thalheim y Wang: *“In general data migration is the process of moving data from legacy data sources of a legacy system into new data sources of a target system, in which legacy and new systems have different data structures.”* (Thalheim y Wang, 2013:260). Es decir, la dificultad del proceso reside no sólo en la transferencia de los contenidos, sino en que los lugares de origen y destino suelen poseer diferentes estructuras, las cuales no siempre son compatibles. Posibilitar esta compatibilidad es, por tanto, el primer reto a alcanzar.

Sin embargo, en otras ocasiones la finalidad de la migración de datos puede responder a otro tipo de motivaciones radicalmente diferentes a las expuestas, si bien más cercanas al tratamiento del lenguaje, como por ejemplo la preservación de lenguas amenazadas o menos descritas, cuya supervivencia puede depender del mantenimiento de repositorios digitales de contenidos en dicha lengua (Beermann y Mihaylov, 2014).

En lo que se refiere a sistemas de PLN, la migración de datos supone una enorme ventaja práctica con respecto a la dotación de contenidos de manera manual, ya que es indudable que la reutilización, con el fin de emprender nuevas tareas, de una colección de datos que ya esté clasificada de manera ordenada, siempre supone una

ventaja. El presente trabajo tiene como hilo conductor esta premisa, es decir, cómo se puede posibilitar la migración de datos desde una fuente a otra para obtener, finalmente, una herramienta enriquecida que contribuya a la resolución de problemas lingüísticos del PLN.

En el actual trabajo, el destino concreto al cual realizamos la proyección de datos se denomina FunGramKB. FunGramKB² (Periñán Pascual y Arcas Túnez, 2004, 2005, 2007a, 2007b, 2010a, 2010b) es una base de conocimiento multilingüe y multipropósito para la resolución de diversas tareas de PLN. Esta base de conocimiento posee diferentes módulos que más adelante definimos en profundidad y que pueden enriquecerse mediante la incorporación de datos procedentes de otras fuentes. En concreto, el presente trabajo se centra en el proceso y resultado de la migración de datos desde fuentes externas accesibles en la web a uno de los módulos de la base de conocimiento FunGramKB, módulo que se denomina *Onomasticón*, el cual debe dotarse de contenidos, proceso que denominamos *población*. Este módulo de la base es el lugar donde se almacena el conocimiento acerca de entidades nombradas (p. ej. personas, lugares, obras artísticas, organizaciones, etc.), contenido al que nos referimos de manera intercambiable como *conocimiento enciclopédico* o *conocimiento cultural* en el presente trabajo. En contraposición a este tipo de conocimiento, utilizamos la denominación de *conocimiento semántico* para aludir al conocimiento conceptual que las palabras denotan, p. ej. los significados de las palabras que podemos encontrar en un diccionario. La combinación de ambos tipos de conocimiento (i.e. conocimiento enciclopédico o cultural y conocimiento semántico) es a lo que llamamos de forma global *conocimiento del mundo*.

² <http://www.fungramkb.com>

La migración de datos al módulo Onomasticón de la base de conocimiento FunGramKB se realiza particularmente desde una fuente concreta, la base de datos DBpedia³ (Auer *et al.*, 2007; Bizer *et al.*, 2009). Esta proyección de los contenidos de DBpedia en la base de conocimiento FunGramKB es de gran importancia, como más adelante especificamos en profundidad, ya que posibilita una mejor resolución por parte del sistema de diversos problemas lingüísticos que surgen durante la realización de tareas de PLN, como puede ser, p. ej., la resolución de la metáfora y la metonimia o la desambiguación de la anáfora, entre otros.

Sin embargo, para que se pueda llevar a cabo el trasvase de datos al módulo Onomasticón de FunGramKB, primero es necesario establecer una metodología que posibilite este proceso. Puesto que se trata de una migración de datos de gran volumen, no sería eficiente realizarla mediante procesos enteramente manuales. Por tanto, en el presente trabajo desarrollamos una metodología semi-automática de población, es decir, que combina fases asistidas con fases automáticas, motivo por el cual merece el calificativo de *semi-automática*. Entendemos por *asistido* aquel proceso que no se realiza de forma íntegramente manual por parte del usuario, sino en el que se le proporciona la asistencia y apoyo de herramientas como, por ejemplo, una interfaz de usuario, un validador de sintaxis, una herramienta de búsqueda de recursos, etc. integradas en la aplicación en la que se trabaja. Por el contrario, un proceso puramente manual no contaría con este tipo de apoyos y se realizaría de forma completa por el humano, sin contar con la ayuda de plataformas, herramientas ni aplicaciones de apoyo.

Partiendo de los planteamientos arriba mencionados, a continuación exponemos los objetivos concretos de esta tesis doctoral.

³ <http://dbpedia.org>

1.3. OBJETIVOS Y ESTRUCTURA DE LA PRESENTE TESIS DOCTORAL

El presente trabajo se estructura en tres objetivos concretos. El primero de ellos consiste en **detallar una metodología de población semi-automática del módulo Onomasticón de FunGramKB**. El Onomasticón de FunGramKB es el módulo donde se almacena el conocimiento cultural, es decir, el conocimiento enciclopédico acerca de entidades nombradas. A través del desarrollo de una metodología de población semi-automática es posible efectuar la migración de este tipo de conocimiento desde las fuentes en la web seleccionadas para ello (i.e. DBpedia) hasta la base de conocimiento FunGramKB. Este trasvase supone un enriquecimiento de FunGramKB, puesto que permite poblar la base de conocimiento con una gran cantidad de información cultural que ayuda a efectuar un tratamiento del lenguaje más rico, ya que posibilita la comprensión por parte de la máquina del comportamiento y descripción de numerosas entidades nombradas del mundo real.

A continuación, tras la implementación de la metodología descrita en nuestro primer objetivo, obtenemos como resultado una serie de estructuras formales denominadas *reglas*. Estas reglas son esquemas basados en plantillas, las cuales permiten la proyección conceptual de los contenidos de DBpedia a la base de conocimiento FunGramKB. Las reglas elaboradas representan constructos conceptuales que codifican una dimensión de una entidad, entendida como una característica o propiedad de dicha entidad que constituye un rasgo definitorio de la misma. Por ejemplo, si se trata de una entidad que es un lugar, una regla conceptual será capaz de reflejar alguna propiedad de este lugar como puede ser su extensión, su altitud, su número de habitantes, el país al que pertenece, etc. Mediante la comparación y análisis de todas las reglas elaboradas, podemos determinar si una estructura formal similar en

las reglas que definen a una entidad, con respecto a las reglas que definen a otra, puede indicar una motivación semántica compartida en dichas propiedades definitorias. Por tanto, nuestro segundo objetivo es **analizar los resultados de la implementación de la metodología descrita, con objeto de confirmar o rebatir la hipótesis de que una estructura formal similar en los constructos elaborados puede indicar una motivación semántica compartida.**

Finalmente, después de elaborar las reglas mencionadas en el objetivo primero y de analizarlas en nuestro segundo objetivo, podemos proceder a determinar si su incorporación a la base de conocimiento FunGramKB supone alguna mejora de la misma para resolver problemas del PLN, en concreto problemas de tipo lingüístico. Los problemas en los que centramos nuestra discusión son la referencia y la correferencia, la anáfora, la metáfora, la metonimia y la *metaftonimia*, de tal forma que nuestro tercer objetivo es, por tanto, **determinar la aplicación que pueden tener los resultados obtenidos, gracias a la metodología implementada, para resolver estos problemas lingüísticos que acucian al PLN.**

Con respecto a la estructura formal de la presente tesis doctoral, en el Capítulo II presentamos diversos modos y teorías sobre la creación y existencia de grandes repositorios digitales de conocimiento, jerarquizados y con un conjunto de reglas e inferencias que relacionen sus conceptos, lo que se denominan ontologías y bases de conocimiento. Asimismo, tratamos de definir con claridad qué son y en qué se diferencian ambos modelos de conocimiento, además de mostrar un retrato del estado actual de la técnica con respecto a ambos. En esta tarea prestamos especial atención a aquellos recursos que puedan demostrar incluir conocimiento conceptual o semántico, así como a otros proyectos que puedan estar vinculados, de uno u otro modo, con la

base de conocimiento FunGramKB y cómo podemos aprovechar de la manera más óptima dicha interacción.

A continuación, en el Capítulo III realizamos una descripción pormenorizada de la base de conocimiento FunGramKB: cuáles son los módulos que la componen, cuál es el modelo lingüístico y teórico sobre el que se sustenta, así como otras consideraciones acerca de sus funcionalidades. De igual forma, analizamos las semejanzas y diferencias que presenta FunGramKB con respecto a otras herramientas similares, así como examinamos las posibilidades de integración de las mismas en nuestra base de conocimiento. En relación a los módulos que componen FunGramKB, centramos nuestra discusión especialmente en el módulo denominado Onomasticón, ya que es en ese módulo donde se lleva a cabo nuestro trabajo experimental y gracias al cual podemos plantearnos los tres objetivos enumerados para esta tesis doctoral.

Continuando con la estructuración de nuestro trabajo, el Capítulo IV lleva como principal hilo conductor la presentación detallada de la metodología desarrollada para poblar con contenidos el módulo Onomasticón de FunGramKB, lo que responde al primero de los objetivos arriba especificados. Se trata de una metodología semi-automática de creación de reglas que permiten el trasvase automático de información procedente de otras fuentes, como es DBpedia, a la base de conocimiento FunGramKB. El inventario completo de las reglas creadas se encuentra en el Apéndice I.

Una vez desarrolladas dichas reglas, en el Capítulo V se plantea un análisis y clasificación de las mismas atendiendo a diversos criterios, así como un examen de las tipologías enunciadas. Esta clasificación tiene por objeto acometer nuestro segundo objetivo, es decir, contribuir a esclarecer la hipótesis inicial de la cual partimos, que enuncia que una similitud formal en la estructura de las reglas puede ser el reflejo de

una motivación semántica compartida entre los constructos conceptuales que éstas representan.

A continuación, el Capítulo VI se orienta hacia el tercero de nuestros objetivos, ya que muestra el impacto y aplicaciones que la población del Onomasticón puede tener sobre problemas recurrentes del PLN como son la referencia, la correferencia, la anáfora, la metáfora, la metonimia y la especial combinación que es la *metaftonimia* (Goosens, 1990). Mediante los ejemplos mostrados en este capítulo vemos cómo el enriquecimiento de FunGramKB a través de la población del Onomasticón supone una mejora de las capacidades del sistema para lidiar con los mencionados problemas lingüísticos.

Finalmente, en el Capítulo VII realizamos un resumen de los contenidos enunciados a lo largo de este trabajo y de las conclusiones que se desprenden de ellos, así como un análisis de las futuras posibilidades, interrogantes y caminos que se abren en la investigación al término del actual trabajo.

CAPÍTULO II

ONTOLOGÍAS Y BASES DE CONOCIMIENTO

2.1. INTRODUCCIÓN

En el presente capítulo efectuamos una radiografía del estado actual de la técnica con respecto a ontologías y bases de conocimiento, centrandó nuestra atención en especial sobre aquellas que, de algún modo, incluyen conocimiento semántico o conceptual y no sólo taxonómico o jerárquico. Asimismo, enumeramos los diferentes enfoques que se pueden adoptar frente a la creación de dichos modelos de conocimiento, principalmente en lo que se refiere al apoyo que éstos realizan sobre una teoría lingüística sólida. No obstante, con carácter previo a la enumeración de dichos modelos de conocimiento y al análisis de sus características, efectuamos ciertas puntualizaciones que contextualizan con más detalle a qué tipo de modelos nos referimos cuando mencionamos una ontología o una base de conocimiento ya que, si bien en algunos círculos donde no es necesaria una escrupulosa especialización técnica ambos términos pueden terminar siendo intercambiables en el lenguaje natural, no lo es así cuando se efectúa un análisis más minucioso y técnico de sus particularidades.

Comenzaremos nuestra descripción mediante la presentación de un ejemplo precursor de estas tareas de recopilación de información universal: el titánico proyecto

Cyc (Lenat *et al.*, 1990), que comenzó a finales de 1984 y que todavía sigue en ejecución. Actualmente, según datos de Cycorp⁴, la base de datos de Cyc contiene más de 500.000 términos, incluyendo unos 26.000 tipos de relaciones y casi 5 millones de hechos y aserciones que relacionan estos términos. El calificativo de *titánico* tiene que ver con el hecho de que la mayoría de estos datos han sido codificados e insertados manualmente en la base de datos de Cyc.

Merece la pena mencionar el punto de partida de este proyecto: ya en 1984, los investigadores de la empresa norteamericana Microelectronics and Computer Technology Corporation (MCC) en Austin, Texas, habiéndose percatado de que las máquinas eran capaces de resolver complejas operaciones de diversos tipos, advirtieron que éstas carecían de respuestas para otras cuestiones mucho más básicas. A este fenómeno lo denominaron “[...] *brittle performance: when the system is confronted by some unanticipated situation, the program is likely to reach the wrong conclusion.*” (Lenat *et al.*, 1990: 32). Esto se puede explicar como una incapacidad de inferir datos, por simples que sean, ya que si no se han anticipado antes, la computadora ignora su existencia. O dicho de otro modo, “*The system breaks down because it cannot detect what, to humans, are very obvious contradictions.*” (Panton *et al.*, 2006:2). Desde el primer momento, se puede advertir que las máquinas carecen de sentido de la “obviedad”, de la capacidad de inferir o de aprender a través de la experiencia, algo que los humanos conocemos como *sentido común*. El objetivo sería, por tanto, tratar de dotar a la máquina de sentido común para que fuese capaz de superar esa fragilidad (“*brittleness*”, arriba mencionada) y ser capaz de razonar. La cuestión es, no obstante, cómo se puede dotar de sentido común a algo que no es humano, cuya capacidad de cognición no viene concedida de manera natural, ni parece que sea capaz de poder

⁴ <http://www.cyc.com>

evolucionar sin la contribución del ser humano. He ahí, precisamente, la respuesta por la que muchos de los investigadores de este campo han abogado: el ser humano es el único que puede otorgar a la máquina ese sentido común⁵. Solamente dotando de sentido común a la máquina, ésta será capaz de resolver cuestiones basadas en el conocimiento semántico de la consulta: la máquina ya no devolverá tan sólo secuencias de letras, sino respuestas. Para lograr este objetivo, un gran número de los intentos de dotación de contenido semántico o de *sentido común* a las máquinas se ha vinculado a la creación de grandes repositorios de conocimiento del mundo, en un lenguaje formal que la máquina pueda entender y procesar, lo que llamamos *ontologías* y *bases de conocimiento* y cuyas diferencias mutuas discutimos a continuación.

2.2. ONTOLOGÍA VS. BASE DE CONOCIMIENTO: DEFINICIONES

Posiblemente existen tantas definiciones de **ontología** como autores trabajando en este campo de estudio, lo que muchas veces provoca que algunas definiciones, cuando se posicionan frente a otras, resulten no sólo disímiles sino, incluso, contradictorias. Asimismo, encontramos que son muchos los factores que pueden afectar a la creación de una ontología y, por ende, a su posterior definición, ostentando incluso los factores personales un papel mayor del que en un principio se pudiera suponer. Con carácter general, la elaboración de una ontología constituye un trabajo colaborativo el cual suele involucrar a un número relativamente elevado de participantes, por lo que alcanzar la homogeneidad en la tarea realizada por ellos puede tornarse, en ocasiones, en una compleja labor, especialmente si la carga subjetiva de cada participante supera a la sistematicidad y protocolo detallado para la creación del modelo. A este respecto, encontramos en el trabajo de Strohmaier *et al.* (2013) un

⁵ No obstante, algunos autores abogan por la capacidad de auto-aprendizaje que las herramientas computacionales de corte ontológico podrán presentar en un futuro, de manera similar a la evolución presente en las entidades vivas, una postura recogida en Fernández *et al.* (2011).

análisis detallado de los factores sociales que pueden influir en la construcción de una ontología, derivados de aspectos tan simples como la distribución del trabajo dentro de la misma o de la manera de trabajar de cada colaborador (p. ej. la intensidad de su actividad, el número de cambios o acciones realizadas en el sistema, etc.).

La mayor parte de las veces, la definición que cada autor realiza del término ontología está íntimamente relacionada con la ontología en sí que éste haya creado o en la que esté trabajando, por lo que, de algún modo, se podría concluir que hay tantas definiciones posibles como ontologías existen, lo que nos impediría listar la totalidad de ellas en este trabajo por obvios motivos de espacio. No obstante, algo que parece común es que tanto las ontologías como las bases de conocimiento pretenden aglutinar información relevante acerca de un ámbito o dominio del mundo, las entidades que lo forman, las características que poseen estas entidades, los eventos que ocurren y las relaciones que tienen lugar entre todos estos conceptos. A modo de comienzo y en relación con este planteamiento, podríamos citar a Noy y McGuinness (2001), quienes expresan que:

*“An **ontology** is a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called **concepts**)), properties of each concept describing various features and attributes of the concept (**slots** (sometimes called **roles** or **properties**)), and restrictions on slots (**facets** (sometimes called **role restrictions**)).”*

Noy y McGuinness (2001:3).

Otra definición previa podemos encontrarla en Uschold y Gruninger: *“Ontology is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve [...] problems [...].”* (Uschold y Gruninger, 1996:103). Quizás lo más interesante de esta definición sea la mención expresa a un fenómeno que más adelante tratamos de manera más extensa en el Capítulo

VI y que consideramos que debería ser el fin ideal de toda herramienta, sea cual sea su naturaleza: el objetivo de resolver problemas (i.e. “*to solve problems*”).

Una tercera definición que se puede presentar, entre las numerosas existentes, es la encontrada en Bateman (1991). Se trata de una definición bastante inclusiva, ya que define mostrando una lista de posibles funciones que una ontología puede o debe cumplir:

“The following list gives an idea of the range of functions adopted in NLP. Ontologies are often expected to fulfill at least one (and often more) of:

- *organizing ‘world knowledge’,*
- *organizing the world itself,*
- *organizing ‘meaning’ or ‘semantics’ of natural language expressions,*
- *providing an interface between system external components, domain models, etc. and NLP linguistic components,*
- *ensuring expressability of input expressions,*
- *offering an interlingua for machine translation,*
- *supporting the construction of ‘conceptual dictionaries’.*

Moreover, an ontology is seen as a very general organizational device: i.e., one that provides a classification system for whatever area of application the ontology is applied to. The organizational resource offered by an ontology has to be re-usable.”

(Bateman, 1991:51).

Esta última idea, la que alude a la reutilización de recursos ontológicos ya existentes, es una valiosa cualidad que más adelante retomamos, puesto que en el actual trabajo tratamos de mostrar los beneficios de esta tarea a través del aprovechamiento que la base de conocimiento FunGramKB efectúa de la migración de datos procedentes de la base de conocimiento DBpedia.

Retornando a las definiciones de ontología, Stumme *et al.* (2006) ponen de manifiesto una definición tan genérica como inclusiva, ya que a pesar de reconocer

todas las discrepancias que pueden existir entre los diversos modos de construir una ontología, admiten que todas comparten ciertas características comunes: “*However, most of them have a certain understanding in common, as most of them include a set of concepts, a hierarchy on them, and relations between concepts. Most of them also include axioms in some specific logic.*” (Stumme *et al.*, 2006:127). No obstante, los mismos autores refinan a continuación mucho más esta definición mediante notación precisa:

“Definition 1. *A core ontology with axioms is a structure $O := (C, \leq_C, R, \sigma, \leq_R, A)$ consisting of*

- *two disjoint sets C and R whose elements are called concept identifiers and relation identifiers, resp.,*
- *a partial order \leq_C on C , called concept hierarchy or taxonomy,*
- *a function $\sigma : R \rightarrow C^+$ called signature (where C^+ is the set of all finite tuples of elements in C),*
- *a partial order \leq_R on R , called relation hierarchy, where $r_1 \leq_R r_2$ implies $|\sigma(r_1)| = |\sigma(r_2)|$ and $\pi_i(\sigma(r_1)) \leq_C \pi_i(\sigma(r_2))$, for each $1 \leq i \leq |\sigma(r_1)|$, with π_i being the projection on the i th component, and*
- *a set A of logical axioms in some logical language L .”*

(Stumme *et al.*, 2006:127).

Así pues, y dejando a un lado por el momento las notaciones estrictamente lógicas o científicas para regresar a descripciones textuales o narrativas de las ontologías, lo que parece ser compartido por muchas de las definiciones existentes en la literatura es que una **ontología** puede definirse como un modelo de conocimiento organizativo, una estructuración jerárquica de clases, conceptos o categorías, que han de estar representados semánticamente de manera formal para que la máquina sea capaz de almacenarlos y tratarlos. El lenguaje de representación que se utiliza para codificar esta información difiere entre las diversas ontologías y bases de conocimiento, lo que puede resultar en un principio un impedimento a la reutilización de las mismas y a la

migración de sus contenidos, a menos que se desarrollen los mecanismos necesarios para salvar dicho obstáculo.

Otra puntualización terminológica que es preciso reseñar es que no se ha de confundir una ontología con una taxonomía léxica, la cual consta de una clasificación de elementos léxicos (i.e. palabras) relacionados. A este respecto, es fundamental remarcar la diferencia entre un concepto y una palabra, del mismo modo que Saussure (1945) hablaba de *significante* y *significado* o Frege (1948) distinguía entre el sentido y la referencia (*“Sinn und Bedeutung”*). Un concepto es una representación mental del estado de las cosas en el mundo o en la mente, similar para todos los seres humanos o, en palabras de Guarino y Giaretta (1995:30), *“conceptualization: an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality”*. Por contra, las palabras constituyen representaciones o instanciaciones léxicas de los conceptos, las cuales varían de una lengua a otra. De este modo, Gruber (1993:199) indica que *“an ontology is an explicit specification of a conceptualization”*, lo que se ha convertido en la definición de ontología más citada en la literatura según Corcho *et al.* (2001), aunque los mismos autores reconocen que otras definiciones son igualmente válidas, como por ejemplo *“an ontology is a formal theory that constrains the possible conceptualisations of the world.”* (Corcho *et al.*, 2001:10). No obstante, nótese que la noción de conceptualización está presente en ambas definiciones, lo que pone de manifiesto la relevancia de la misma. Finalmente, una definición que parece sencilla a la vez que clarificadora es la que nos aporta el World Wide Web Consortium (W3C): *“An ontology defines the terms used to describe and represent an area of knowledge.”* (Heflin, 2004: sección 1.1).

Por otro lado, una **base de conocimiento** se puede definir como una base de datos dinámica, representada formalmente para que sea tratable por la máquina y que ha

de estar dotada de un motor de razonamiento por inferencia. Esta herramienta llamada *motor de razonamiento* sería la responsable del nivel heurístico del sistema, es decir, de razonar de manera eficiente con aquellos datos que el sistema conoce gracias a su previa introducción manual, asistida o automática. No obstante, al igual que ocurre con la noción de ontología, podemos encontrar toda una plétora de definiciones para el término *base de conocimiento* que, de nuevo, suelen estar vinculadas al trabajo particular de cada autor con dichos modelos de conocimiento. En Corcho *et al.* (2001) se pone de manifiesto este hecho del siguiente modo: *“Other definitions emerge as a consequence of how their authors build and use ontologies. We can distinguish here between top-down or bottom-up approaches depending on what was first the ontology or the KB [Knowledge Base].”* (Corcho *et al.*, 2001:10). Con objeto de mantener el paralelismo con los autores anteriormente citados, encontramos en Noy y McGuinness (2001) la siguiente referencia, que enlaza con su definición de ontología y la mención de “clases”: *“We can then create a knowledge base by defining individual instances of these classes filling in specific slot value information and additional slot restrictions.”* (Noy y McGuinness, 2001:3).

Puesto que, como se ha observado, la tendencia es que cada autor defina una base de conocimiento de la manera más coherente con su metodología de trabajo, esto conlleva que diferentes autores entiendan el proceso de generación y construcción de ontologías y bases de conocimiento de maneras inversas: la ontología puede ser el punto de partida sobre el que la base de conocimiento se construye, se sostiene y de donde se destilan los datos necesarios para el proceso de razonamiento; o bien la ontología puede ser un producto derivado de la depuración de elementos pre-existentes en la base de conocimiento. Esta última opción, si bien puede resultar contradictoria para algunos autores, es mencionada por Corcho *et al.*: *“The opposite approach was taken at the*

KACTUS project [Bernaras et al., 96], where the ontology is built after a process of abstraction of the content already represented in a knowledge base.” (Corcho et al., 2001:10).

El hecho de decantarse por una u otra opción no dependerá más que del tipo de modelo que se esté perfilando y de las funciones que se persiga cumplir mediante su uso. Sin embargo, se podría argumentar que, a pesar de que esta creación de recursos *ad hoc* podría resultar eficiente para el modelo en particular que se esté utilizando, dicho planteamiento va en detrimento de la deseada reusabilidad de la misma en diferentes contextos o dominios (Periñán Pascual y Arcas Túnez, 2006; Mairal Usón y Periñán Pascual, 2009), lo que podría catalogarse como una creación de contenidos poco eficiente o, simplemente, de un inconveniente de dicha acción. Sirva como resumen de esta problemática lo que se manifiesta en Ding *et al.*: *“Failing to find a proper ontology always leads to the creation of a new ontology, which is often too customized to be reused.”* (Ding *et al.*, 2004:652), lo que plantea un serio problema a la hora de posibilitar la migración de datos de un recurso a otro.

2.3. LA MIGRACIÓN DE DATOS EN LAS ONTOLOGÍAS Y BASES DE CONOCIMIENTO

Una vez que hemos mostrado las principales características que definen tanto a ontologías como a bases de conocimiento, podemos trasladar nuestra discusión nuevamente al tema que introducíamos en el capítulo anterior y que se enfocaba al traspaso de datos entre diversas fuentes y a cómo pueden reutilizarse los datos contenidos en cada una de ellas mediante la migración entre recursos. En concreto, es pertinente mencionar uno de los campos donde actualmente se desarrolla un gran número de proyectos que buscan posibilitar la proyección de datos desde ontologías y bases de conocimiento: la Web Semántica.

2.3.1. LA MIGRACIÓN DE DATOS PARA LA WEB SEMÁNTICA

Uno de los campos en los que actualmente se lleva a cabo un gran volumen de migración de datos es la Web Semántica. La relación que ésta guarda con los modelos de conocimiento que presentamos en el actual capítulo (i.e. ontologías y bases de conocimiento) reside en que muchos de estos recursos encuentran una aplicación práctica de su utilidad en diversos sistemas relacionados con ella, ya que la creación de estos cuerpos de conocimiento constituye un paso previo necesario para que los agentes de Web Semántica puedan desarrollarse y llevar a cabo los procesos de razonamiento automático que sean pertinentes: *“For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning.”* (Berners-Lee *et al.*, 2001:5). Por tanto, encontramos que las tecnologías de Web Semántica deben reposar sobre una base ontológica puesto que, en definitiva, estas tecnologías no albergan unos objetivos demasiado diferentes a los de muchas ontologías y bases de conocimiento: *“‘Expressing meaning’ is the main task of the Semantic Web”*. (Stojanovic *et al.*, 2002:1100).

Si se desea comenzar mediante la aportación de una definición algo más extensa acerca de qué es la Web Semántica y para qué sirve, encontramos la siguiente definición en Berners- Lee *et al.*:

“The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The first steps in weaving the Semantic Web into the structure of the existing Web are already under way. In the near future, these developments will usher in significant new functionality as machines become much better able to process and “understand” the data that they merely display at present.”

(Berners-Lee *et al.*, 2001:3-4).

La Web Semántica, por tanto, no sólo intenta ser capaz de proporcionar respuestas concretas a las consultas explícitas realizadas de manera textual por los

usuarios, sino que pretende ir más allá, en un intento por transformar a las computadoras, tal y como las conocemos, en *máquinas sociales* (Hendler y Berners-Lee, 2010). Entendida la Web Semántica como una herramienta al servicio de los humanos, la necesidad de efectuar una pregunta de manera explícita se puede ver obviada, de tal forma que la integración de sistemas de Web Semántica en nuestra vida diaria sea capaz de resolver problemas cotidianos no sólo previa consulta manifiesta formulada en lenguaje natural por el usuario, frente a la pantalla de un ordenador o dispositivo de interfaz humano-máquina (p. ej. una tableta, un *smartphone*, etc.) sino en otras máquinas que utilizamos a diario (p. ej. electrodomésticos), lo que Berners-Lee *et al.* (2001) resumen de esta manera tan gráfica:

“Initially it will let cell phones and other nonstandard Web clients describe their characteristics so that Web content can be tailored for them on the fly. Later, when we add the full versatility of languages for handling ontologies and logic, devices could automatically seek out and employ services and other devices for added information or functionality. It is not hard to imagine your Web-enabled microwave oven consulting the frozen-food manufacturer's Web site for optimal cooking parameters.”

(Berners-Lee *et al.*, 2001:16).

Actualmente, uno de los proyectos más populares relacionados con la Web Semántica e impulsado por Berners-Lee es el proyecto *Linking Open Data*⁶ (Berners-Lee y O'Hara, 2013), que desde el año 2007 realiza un seguimiento y control de los contenidos y datos de la llamada *Linked Data Web*, lo que supone un ejercicio de migración de datos a gran escala. La idea principal que subyace en este proyecto es que la Web no sea solamente un medio de publicación de contenidos, sino algo más: una Web que permita tanto leer como redactar contenidos. Esta concepción colaborativa de la Web, lejos de ser una idea relativamente reciente, con el advenimiento en los últimos años de múltiples páginas de publicación de contenidos, redes sociales, blogs, *wikis*,

⁶ <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

etc., responde a la idea inicial que los creadores de la WWW albergaban con respecto a ella, la cual los autores desvelan de esta forma:

“When the Web was created, the idea was for a read–write Web. The original browser was actually an editor; if you had the right access, you could change the text or add links to other places on the Web. The idea of hypertext trails was an extremely important idea to facilitate collaboration [...], but the Web took off primarily as a publishing medium. Now there are more outlets for writers—one can write a blog or contribute to a wiki or OpenStreetMap—but there is a need for more modalities for interactivity. If the Web is genuinely to be a read–write Web, and if we are to pursue a Web of Data, then it is essential that the Data Web should not be read-only.”

(Berners-Lee y O’Hara, 2013:2).

La *Linked Data Web* recibe contenidos migrados desde diversos orígenes, no sólo desde entidades comerciales como Google⁷, Yahoo!⁸, Bing⁹ o desde entidades y agencias gubernamentales, sino que uno de los primeros en contribuir a este proyecto fue DBpedia, una plataforma de gran importancia para el actual trabajo y la cual analizamos detalladamente más adelante, especialmente en relación a su conexión directa con FunGramKB, la base de conocimiento en la que se desarrolla nuestra investigación y que describimos en profundidad en el siguiente capítulo.

Desde una perspectiva diacrónica, no son pocos los intentos que se han realizado desde la creación de la WWW y, en especial, en épocas más recientes, para reutilizar de manera semántica la información contenida en textos escritos en lenguaje natural o para trabajar en la recuperación de estos contenidos, posibilitando su migración entre diversos modelos de conocimiento. En especial, muchos de estos sistemas toman la forma de motores de búsqueda, los cuales podemos identificar como un excelente modelo de migración de datos entre diversas fuentes.

⁷ <http://www.google.com>

⁸ <http://www.yahoo.com>

⁹ <http://www.bing.com>

Por ejemplo, uno de los motores de búsqueda que podemos comenzar mencionando es el sistema *Serene - Semantic Web search engine* (Fazzinga et al., 2011), que permite procesar semánticamente búsquedas Web, además de evaluar consultas Web complejas que necesitan utilizar razonamiento en la red, para lo cual asumen que debe existir una ontología subyacente. Es frecuente que en este tipo de iniciativas se efectúe una prueba de concepto en servidores o páginas web reales, con objeto de conferir validez a la herramienta y demostrar el impacto real de las funcionalidades que se ofrecen. En el caso de *Serene*, el campo de pruebas en el que se ha realizado su aplicación es la base de datos de películas *Internet Movie Data Base* (IMDB¹⁰), mediante la implementación de una interfaz de búsqueda web semántica para las páginas de dicho sitio web, haciéndolo posible a pesar de que estas páginas no están anotadas semánticamente.

Otro interesante proyecto es *Swoogle*¹¹ (Ding et al., 2004; Ding et al., 2005), una herramienta creada como motor de búsqueda de información, páginas web o documentos con contenido semántico en formato *Resource Description Framework* (en adelante, RDF) u *Ontology Web Language* (en adelante, OWL)¹². Además de ser capaz de localizar dicho contenido y los denominados *Semantic Web Documents*, *Swoogle* también es capaz de indexar estos documentos mediante el almacenaje de los metadatos de los mismos y, de esta forma, efectuar conexiones o relacionarlos con otros similares en la web. Un *Semantic Web Document* (en adelante, SWD) es una página web que genera una representación almacenable de un gráfico en RDF utilizando alguno de los lenguajes recomendados para la sintaxis de RDF, p. ej. RDF/XML, N-Triples o N3

¹⁰ <http://www.imdb.com>

¹¹ <http://swoogle.umbc.edu>

¹² *Resource Description Framework* (RDF) y *Web Ontology Language* (OWL) son lenguajes de etiquetado para describir e intercambiar metadatos en la Web. RDF permite describir contenidos a partir de las unidades semánticas que los componen, mientras que OWL permite una mayor interoperabilidad, ya que aporta vocabulario adicional junto a una semántica formal.

(Ding *et al.*, 2005). Incluso, *Swoogle* es capaz de determinar el rango de cada SWD, de modo que se mida la importancia de cada documento en la Web Semántica con técnicas que con el tiempo son depuradas y mejoradas (Ding *et al.*, 2005). Si bien este intento parece ser una herramienta tan válida como cualquier otra, encontramos que el principal problema de la misma es que, para que *Swoogle* sea capaz de localizar y tratar el contenido semántico, es necesario que dicho contenido esté etiquetado en los lenguajes RDF u OWL que *Swoogle* es capaz de procesar. A pesar de que estos lenguajes son lo suficientemente populares en la Web Semántica como para resultar prolíficos¹³, siempre serán más restringidos que, obviamente, si la herramienta no exigiera ningún lenguaje determinado. No obstante, debe tenerse en cuenta que el principal objetivo de *Swoogle* es clasificar los contenidos semánticos etiquetados como tales y no la tarea de conferir a contenidos web no semánticos dicha propiedad, por lo que el uso de los lenguajes RDF u OWL entendemos que actúa como elemento facilitador para la localización de dichos contenidos semánticos.

Nuevos proyectos más recientes relacionados con la proyección de datos en la Web Semántica y, en este caso, realizados dentro de un dominio concreto (p. ej. el mercado inmobiliario en Oxford, Reino Unido) son los proyectos LOD2¹⁴ y DIADEM¹⁵, que han dado lugar al desarrollo de la herramienta DEQA¹⁶. En palabras de sus creadores:

“We present DEQA, a system that allows the easy combination of semantic technologies, data extraction, and natural language processing and demonstrate its ability to answer questions on Oxford’s real estate market. [...] For example a query

¹³ No en vano, ambos lenguajes forman parte de los estándares fijados por el W3C con respecto a la creación de contenidos de Web Semántica, como se puede ver en el siguiente enlace: <http://www.w3.org/standards/semanticweb>.

¹⁴ <http://lod2.eu/Welcome.html>

¹⁵ <http://diadem.cs.ox.ac.uk>. Este proyecto también ha apoyado el desarrollo del sistema *Serene* (Fazzinga *et al.*, 2011), referenciado anteriormente en esta sección.

¹⁶ <http://aksw.org/Projects/DEQA.html>

like “find me a flat to rent close to Oxford University with a garden” can be answered by DEQA. However, this cannot be achieved without adaptation to the specific domain.”

(Lehmann *et al.*, 2012: 132).

Por tanto, el sistema DEQA constituye una interfaz que media entre el contenido no semántico ni jerarquizado que ofrecen los portales inmobiliarios de la ciudad británica de Oxford en lenguaje HTML¹⁷, que es el lenguaje en el que habitualmente están codificadas las páginas web, lo que ya establece un paso hacia la flexibilidad, dado que no se precisa que los documentos de origen estén en lenguajes concretos como RDF u OWL. Asimismo, el sistema permite también la transformación de dicha información al formato RDF para posibilitar su vínculo con otro tipo de conocimiento (p. ej. con datos de ubicación extraídos de *LinkedGeoData*¹⁸) para proporcionar finalmente la respuesta a las preguntas específicas que el usuario pueda formular (Lehmann *et al.*, 2012). El hecho de que la aplicación del sistema DEQA se efectúe y desarrolle en un dominio tan concreto como el mercado inmobiliario en la localidad de Oxford no debe resultar algo que le resta relevancia sino, por el contrario, dicha aplicación puede conferir a la herramienta una validación en tiempo real del impacto que su uso puede tener sobre un problema existente en el mundo real. Este fenómeno, por tanto, demuestra que cualquier aplicación específica de la Web Semántica, debidamente desarrollada y definida, puede servir para mejorar una problemática concreta del mundo a nivel particular, sin perjuicio de que esta aplicación, por el momento local, pueda dar lugar a una extrapolación a otros dominios del conocimiento o lugares de aplicación.

¹⁷ Acrónimo de *HyperText Markup Language*. Se trata de un lenguaje de marcado para la elaboración de páginas web, el cual es un estándar en la red.

¹⁸ <http://linkedgeo.org/About>

Otros trabajos, esta vez aplicables a campos o dominios genéricos¹⁹, proponen combinar las ventajas de las técnicas de Web Semántica con las de minería web (Stumme *et al.*, 2006), con objeto de mejorar tanto los contenidos como el proceso para su obtención, de modo que se resuelva la problemática expuesta a continuación:

“They [the Semantic Web and Web Mining] complement each other well because they each address one part of a new challenge posed by the great success of the current WWW: Most data on the Web are so unstructured that they can only be understood by humans, but the amount of data is so huge that they can only be processed efficiently by machines. The Semantic Web addresses the first part of this challenge by trying to make the data (also) machine-understandable, while Web Mining addresses the second part by (semi-)automatically extracting the useful knowledge hidden in these data, and making it available as an aggregation of manageable proportions.”

(Stumme *et al.*, 2006:124).

La idea presente en este extracto acerca de crear procesos semi-automáticos más eficientes y rigurosos subyace también en el trabajo desarrollado para la presente tesis doctoral. En concreto, en la creación del módulo Onomasticón de la base de conocimiento FunGramKB, en la cual desarrollamos nuestro trabajo, también se realiza un proceso semi-automático para dotar de contenidos a dicho módulo, como detallamos a lo largo de los Capítulos III, IV y V.

Finalmente, otro sistema vinculado muy de cerca con el proyecto *Linked Open Data* anteriormente mencionado es la herramienta SWSE- *Semantic Web Search Engine* (Hogan *et al.*, 2011), que surge como respuesta a un interrogante: “*Assuming large-scale adoption of high-quality RDF publishing on the Web, could a search engine indexing RDF feasibly improve upon current HTML-centric engines?*” (Hogan *et al.*, 2011: 366). Si bien los creadores de SWSE reconocen otros intentos de creación de motores de búsqueda de web semántica, algunos de los cuales ya han sido reseñados en

¹⁹ Si bien el campo de pruebas en el que se ha desarrollado el trabajo expuesto en Stumme *et al.* (2006) es una página web ficticia de turismo y viajes.

la presente sección, como *Serene* o *Swoogle*, las dos particularidades que los autores consideran que confieren a su herramienta una diferencia sobre el resto son el hecho de que el sistema SWSE es, por un lado, capaz de ser ampliado modularmente a un volumen mayor de datos y, por otro, que se trata de un sistema que se mantiene robusto²⁰ frente a posibles interferencias como datos heterogéneos, confusos o contradictorios procedentes de diversos orígenes. Esta última idea guarda relación con el problema que supone evaluar la calidad de la información que podemos encontrar en la Web Semántica, si bien observamos que éste constituye un problema no sólo aplicable a la Web Semántica, sino también a la actual Web que conocemos. Encontramos trabajos como el de Baillie *et al.* (2012) que se centran en evaluar dicha calidad, así como la fiabilidad o grado de confianza de los orígenes y métodos de tratamiento del contenido a proyectar. Para ello, se sirven de redes de sensores de observación y otras herramientas relacionadas. Los sensores de observación son sensores físicos de diversa naturaleza como, p. ej., teléfonos móviles de usuarios, monitores de estados fisiológicos en los hospitales o sensores de conservación medioambiental. En cuanto a las herramientas relacionadas, éstas son principalmente las recomendaciones efectuadas por el grupo de trabajo en la materia, *Provenance Working Group*, del W3C²¹, con el fin de determinar la validez de los recursos. El trabajo de Baillie *et al.* (2012) constituye, más que un sistema de creación y compilación, un sistema de evaluación de la Web Semántica, lo que también resulta una interesante y necesaria vertiente del trabajo realizado en este campo.

Así pues, una vez que hemos descrito algunos de los principales trabajos cuyo objetivo es proyectar datos de diversas fuentes hacia la Web Semántica, en la siguiente

²⁰ Entendemos por *robusto* un sistema que no tiene errores con facilidad y que responde de manera correcta durante la mayor parte de su funcionamiento.

²¹ <http://www.w3.org/2011/prov>

sección trasladamos nuestra discusión a otra característica importante de las ontologías y bases de conocimiento que sirven de origen para dicha proyección de datos. Esta característica que procedemos a describir son los fundamentos lingüísticos sobre los que ontologías y bases de conocimiento pueden sustentarse.

2.4. FUNDAMENTOS LINGÜÍSTICOS DE LAS ONTOLOGÍAS: SEMÁNTICA PROFUNDA VS. ENFOQUE RELACIONAL

Es un hecho conocido que, a lo largo de la historia de la ingeniería lingüística, la lingüística informática o la lingüística computacional²², no todos los modelos de conocimiento o herramientas para el PLN se han provisto de una base lingüística teórica robusta sobre la que apoyarse antes de su creación. Este suceso, si bien ha permitido crear herramientas para el PLN con mayor rapidez y facilidad, puede abocar a éstas a una propensión a los fallos, puesto que carecen de un sistema complejo y fundamentado de comprensión del lenguaje natural (Periñán Pascual y Mairal Usón, 2009). Debido a diversas causas y motivaciones a lo largo de la historia²³, la colaboración entre ingenieros informáticos y lingüistas teóricos no siempre ha sido la que se desearía para producir modelos más sólidos, robustos y fundamentados, lo que ha originado resultados poco deseables, como resume Periñán Pascual:

“En definitiva, las investigaciones actuales en ingeniería lingüística no se fundamentan en la lingüística, sino en la estadística y la teoría de las probabilidades, incluso a costa de obtener soluciones sucias (Ferrari, 2004), i.e. soluciones menos plausibles desde el punto de vista teórico”.

(Periñán Pascual, 2012a:27).

²² La propia definición de las denominaciones de “lingüística computacional”, “lingüística informática” o “ingeniería lingüística” ya supone un reto, como bien apunta Periñán Pascual, quien incluso se refiere a ello como “*un caos terminológico*” (Periñán Pascual, 2012a:15).

²³ En Periñán Pascual (2012a) se efectúa una completa descripción diacrónica de los movimientos y tendencias desde los años 40 y 50 hasta la actualidad.

Sumado a esta problemática inicial o, precisamente, quizás siendo ésta el origen, otro aspecto relevante a tratar es la profundidad semántica con la que es representado el conocimiento que se incluye en una ontología o una base de conocimiento. A la hora de representar el conocimiento en sistemas de PLN, es necesario elegir entre dos tipos de significado: el significado relacional, que pertenece a la semántica superficial o *surface semantics* y el significado conceptual, perteneciente a la semántica profunda o *deep semantics*. El primero es aquel que podemos expresar mediante la asociación entre diferentes unidades léxicas, muy en la línea de lo que denominábamos taxonomías léxicas anteriormente, mientras que el segundo alude a la descripción mediante rasgos semánticos o *primitivos* del contenido de cada unidad. Por consiguiente, el significado relacional no es puramente descriptivo, ya que se limita a señalar las conexiones entre unidades léxicas, sin ahondar en su contenido semántico individual (Mairal Usón y Perriñán Pascual, 2010a), al contrario que el significado conceptual. De manera ilustrativa, se pueden citar modelos como EuroWordNet²⁴, que utilizan un enfoque relacional²⁵. Si se detalla un poco más la profundidad con la que puede realizarse la descripción de la semántica léxica y la granularidad con la que se define el contenido cognitivo, podemos incluso desdoblar el enfoque *no-profundo* en dos sub-enfoques: el enfoque superficial y el enfoque intermedio, según la tipología propuesta por Perriñán Pascual y Carrión Varela (2011):

- “ (i) *enfoque superficial, por medio de asociaciones semánticas con otras unidades léxicas,*
- (ii) *enfoque intermedio, por medio de relaciones conceptuales binarias formando una simple matriz de atributo-valor, o*

²⁴ <http://www.illc.uva.nl/EuroWordNet>

²⁵ Perriñán Pascual y Arcas Túnez (2007b) realizan una detallada comparación entre FunGramKB y EuroWordNet y las carencias que un enfoque relacional supone.

(iii) *enfoque profundo, por medio de rasgos semánticos que proporcionen una verdadera definición, p. ej. a través de una representación semántica proposicional.”*

(Periñán Pascual y Carrión Varela, 2011:90).

Así pues, muchos de los modelos de conocimiento creados hasta ahora para sistemas de PLN sienten predilección por enfoques relacionales como los dos primeros (i.e. enfoque superficial e intermedio), puesto que las ventajas que éstos ofrecen a corto plazo son evidentes: relativa sencillez y rapidez de creación²⁶. Sin embargo, estas ventajas pueden resultar a la vez inconvenientes, puesto que dicha sencillez y rapidez se transforman en obstáculos a la reusabilidad o traslación a otros sistemas, en una restricción de poder expresivo y en una alta redundancia (Periñán Pascual y Arcas Túnez, 2007b; Mairal Usón y Periñán Pascual, 2010a). No obstante, pese a los aparentes impedimentos iniciales de adoptar un enfoque de semántica profunda, ya que se requiere un gran esfuerzo en tiempo y especialización²⁷, su utilización provoca que la alta redundancia y la restricción de poder expresivo se vean minimizadas significativamente, mientras que la reusabilidad se ve potenciada.

En relación a los problemas de redundancia en bases de conocimiento relacionales, Periñán Pascual y Arcas Túnez (2007b) subrayan que la redundancia que el multilingüismo provoca en bases como EuroWordNet no tiene lugar en FunGramKB, ya que los postulados de significado (i.e. propiedades conceptuales) presentes en FunGramKB son representaciones cognitivas de conceptos, comunes a todas las lenguas naturales, a las que se asignan las diferentes unidades léxicas que constituyen las lexicalizaciones en cada lengua natural. Esta circunstancia, además de disminuir la

²⁶ Sirva como ejemplo de este fenómeno la cita de Frederick Jelinek referenciada en Periñán Pascual: “*Anytime a linguist leaves the group the recognition rate goes up.*”(Periñán Pascual, 2012a:14).

²⁷ Como ejemplo, no olvidemos el proyecto Cyc mencionado al comienzo de este capítulo.

redundancia, potencia significativamente la informatividad del repositorio de conocimiento semántico de la herramienta (Periñán Pascual y Arcas Túnez, 2007b).

Mairal Usón y Periñán Pascual (2010b) también arguyen que algunas unidades conceptuales son muy difíciles de representar de manera relacional, como por ejemplo REMEMBER, LOVE o FORGET²⁸, por lo que en conjunto se podría decir que, tras sopesar ambas opciones, es más que razonable pensar que un modelado del conocimiento basado en la semántica profunda constituye un esfuerzo eficiente y amortizable.

De manera paralela a la decisión de adoptar un enfoque de semántica profunda, se ha de plantear otra importante cuestión: la necesidad de contar con una teoría lingüística sólida sobre la que se sustente la base de conocimiento, con objeto de dotar a ésta de validez y robustez. Existen sistemas de PLN que, para ser creados, han hecho uso de manera explícita de una teoría lingüística, si bien después se ha constatado que el modelo ontológico sobre el que se han construido ha resultado ser de naturaleza superficial, lo que quizás haya podido mermar la efectividad del propósito inicial de crear una herramienta robusta. Un ejemplo que podemos mencionar a este respecto es el sistema PHORA (Palomar Sanz *et al.*, 2000), el cual, si bien proclama el contar con una base lingüística sólida (el formalismo de las gramáticas Léxico-Funcionales), después utiliza como alimentación ontológica a EuroWordNet, que ellos mismos definen como un diccionario de conjuntos de sinónimos o *synsets* (cf. Palomar Sanz *et al.*, 2000), lo que dista superlativamente de una base de conocimiento de semántica profunda. Además, las restricciones y preferencias en las que se basa el sistema, a pesar de ser obtenidas de diversos tipos de conocimiento (morfológico, semántico, sintáctico, etc.), excluyen explícitamente el conocimiento cultural o enciclopédico. No obstante, no es de

²⁸ En versales porque mediante dichas expresiones nos referimos al concepto que cada una de ellas denota, no a la instanciación léxica de cada concepto.

extrañar el hecho de que esta solución haya sido escogida, ya que muchos otros autores respaldan este enfoque e incluso intentan suplir el conocimiento del mundo con técnicas de aprendizaje automático o *machine learning*, algo que se resume de manera muy ilustrativa en Mitkov (2002):

“*Natural language understanding requires a huge amount of knowledge about morphology, syntax, semantics, discourse and pragmatics and general knowledge about the real world but the encoding of all this knowledge represents an insurmountable impediment for the development of robust NLP systems. As an alternative to knowledge-based systems, machine learning methods offer the promise of automating the acquisition of this knowledge from annotated or unannotated corpora by learning from a set of examples (patterns).*”

(Mitkov, 2002:113).

En relación al modelo lingüístico que actúa como referencia en FunGramKB para los niveles léxico y gramatical, se trata del Modelo Léxico Construccional²⁹, referido como MLC o LCM por sus siglas en inglés, *Lexical Constructional Model* (Ruiz de Mendoza Ibáñez y Mairal Usón, 2008; Mairal Usón y Ruiz de Mendoza Ibáñez, 2009; Mairal Usón *et al.*, 2012; Perrián Pascual y Mairal Usón, 2012). Este modelo constituye una reconciliación entre paradigmas funcionales y cognitivos del lenguaje, lo que supone un enriquecimiento mutuo entre ellos, al dar cabida a diversos niveles y aspectos del significado:

“*The primary concern of the LCM is to develop a usage-based, comprehensive theory of meaning construction that aims to give explanations of how all aspects of meaning, including those that go beyond so-called core-grammar (e.g., traditional implicature, illocutionary force, and discourse coherence) interact among one another.*”

(Ruiz de Mendoza Ibáñez y Mairal Usón, 2008:355).

Este planteamiento es el que, en la práctica, une los postulados *proyeccionistas* de la Gramática del Papel y la Referencia (RRG, por sus siglas en inglés, *Role and*

²⁹ Para más información acerca del MLC, puede visitarse la página web del grupo de investigación Lexicom, <http://www.lexicom.es>. Para una revisión crítica del modelo, referimos a Butler (2009, 2012).

Reference Grammar) de Van Valin y LaPolla o de la Gramática Funcional (*Functional Grammar*, FG) de S.C. Dik, junto a la vertiente *construccionista* de Fillmore y Kay, entre otros (Mairal Usón y Ruiz de Mendoza Ibáñez, 2009). Se trata, por tanto, de combinar perspectivas de modelos funcionales como la RRG y FG mencionadas, junto a los postulados de la Lingüística Cognitiva de Lakoff y la Gramática de Construcciones de Goldberg (Ruiz de Mendoza Ibáñez y Mairal Usón, 2008), con objeto de obtener una teoría sólida, versátil y que permita diferentes aplicaciones para diversas tareas de PLN:

“En esencia, el MLC nos ofrece las herramientas analíticas necesarias para un análisis detallado de las dimensiones proposicional y pragmático-discursiva del significado, lo que permite formular una interpretación semántica plenamente especificada de un texto, que bien podría ser el texto de entrada en un ordenador, o un robot o un sistema inteligente.”

(Mairal Usón, 2012:184).

Así pues, en el actual capítulo hemos descrito las principales diferencias existentes entre ontologías y bases de conocimiento, con motivo de contextualizar el presente trabajo, además de haber enumerado diversos proyectos que utilizan dichas herramientas para efectuar una proyección de datos, especialmente en la Web Semántica. Igualmente, hemos descrito los diferentes enfoques que se pueden utilizar a la hora de crear ontologías y bases de conocimiento (i.e. enfoque superficial, intermedio y profundo), con objeto de ilustrar las principales ventajas e inconvenientes que presenta la utilización de cada uno, así como hemos expuesto la importancia de escoger una teoría lingüística sólida sobre la que basar la construcción de los módulos lingüísticos de una base de conocimiento.

En el Capítulo III iniciaremos la descripción detallada del modelo de conocimiento concreto utilizado para el presente estudio: la base de conocimiento

FunGramKB, junto a otras herramientas que pueden contribuir a la población de dicha base y la forma en que éstas pueden ser aprovechadas y reutilizadas.

CAPÍTULO III

FUNGRAMKB Y EL ONOMASTICÓN

3.1. ¿QUÉ ES FUNGRAMKB?

Con el objetivo de comenzar la descripción exhaustiva de la base de conocimiento FunGramKB³⁰ mediante una definición explícita de la misma, obsérvese el siguiente extracto de Perinián Pascual y Arcas Túnez (2010b):

"FunGramKB Suite is a user-friendly online environment for the semiautomatic construction of a multipurpose lexico-conceptual knowledge base for natural language processing (NLP) systems, and more particularly for natural language understanding."

(Perinián Pascual y Arcas Túnez, 2010b:2667).

En primer lugar, es preciso efectuar una distinción entre FunGramKB Suite, que es el entorno provisto de herramientas para trabajar con FunGramKB y FunGramKB en sí, que es la base de conocimiento resultante. Con respecto a las características de FunGramKB, encontramos en la definición arriba mostrada que se trata de una base de conocimiento **léxico-conceptual** al incorporar componentes léxicos y conceptuales a través de los diferentes módulos que la componen, teniendo así cabida tanto los

³⁰ <http://www.fungramkb.com>

constructos conceptuales como las instanciaciones léxicas de estos constructos, en cada lengua natural de las incluidas en la base de conocimiento.

En segundo lugar, se puede apreciar que FunGramKB se define no como ontología, sino como **base de conocimiento** que integra los niveles léxico y conceptual, y que es **multipropósito** debido a la potencial reusabilidad de este modelo de conocimiento en diversas tareas de PLN. No en vano, se ha de recordar que la noción de reusabilidad se ha mencionado anteriormente como una característica que idealmente han de poseer este tipo de herramientas, con el fin principal de permitir la proyección de los datos contenidos en ellas. La reusabilidad de FunGramKB alude a que se trata de una base de conocimiento, en primer lugar, multifuncional, ya que puede utilizarse en diversos ámbitos, como por ejemplo la creación de agentes o motores de búsqueda inteligentes, traducción automática o diccionarios que posibiliten una búsqueda conceptual (Periñán Pascual y Arcas Túnez, 2006; Mairal Usón y Periñán Pascual, 2010b), así como para otro tipo de aplicaciones que proporcionan respuestas a problemas lingüísticos concretos en el ámbito del PLN, lo cual detallamos de manera más extensa en el Capítulo VI del presente trabajo. Es precisamente en dicho capítulo donde vemos cómo FunGramKB posee una capacidad mejorada para resolver problemas lingüísticos del PLN como la referencia y la correferencia, la anáfora, la metáfora, la metonimia y la *metaftonimia* (Goosens, 1990). En segundo lugar, otra vertiente de la reusabilidad alude a que se trata de una base multilingüe, incluyendo en la actualidad la lexicalización de conceptos en siete lenguas naturales (inglés, español, italiano, francés, alemán, búlgaro y catalán).

FunGramKB está estructurada en tres grandes niveles de información, que a su vez se subdividen en varios módulos independientes pero interrelacionados (Periñán Pascual y Arcas Túnez, 2006; Periñán Pascual y Arcas Túnez, 2010b). Estos tres

grandes niveles son el **nivel léxico** (conocimiento lingüístico), el **nivel gramatical** (conocimiento acerca de esquemas construccionales) y el **nivel conceptual** (conocimiento no lingüístico). Cada nivel se subdivide en varios módulos de la manera que muestran las figuras que se incluyen a continuación:

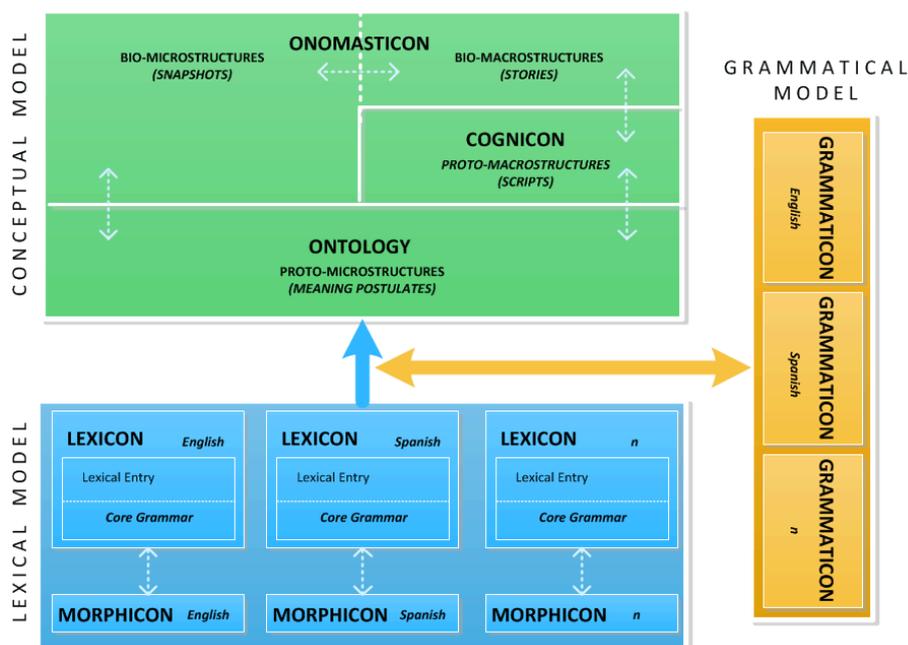


Figura 1. Arquitectura de FunGramKB³¹

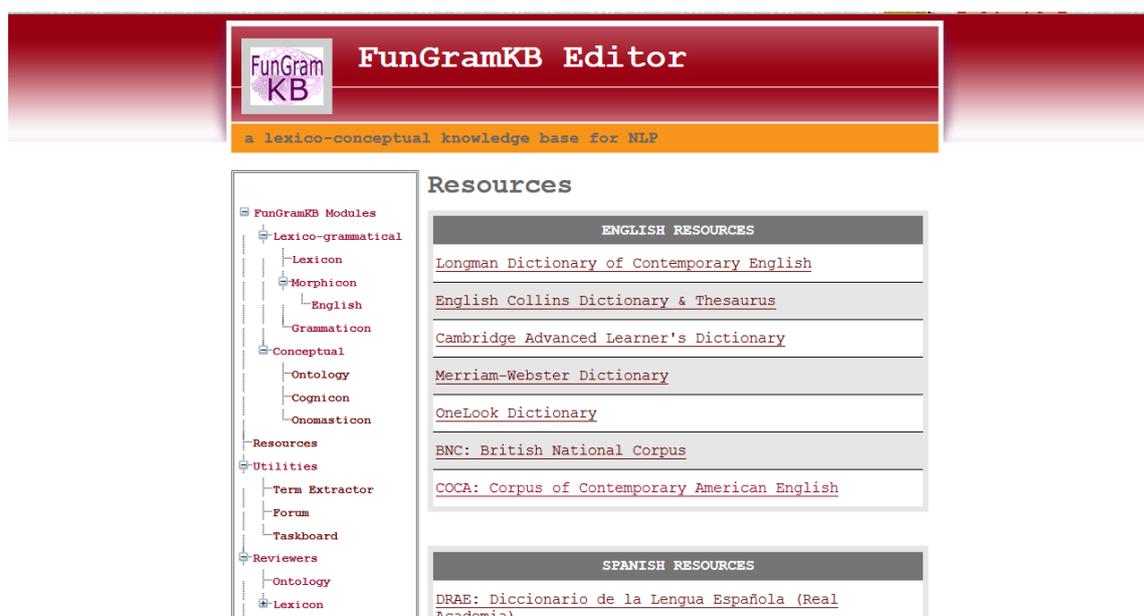


Ilustración 1. Vista parcial de FunGramKB Suite, mostrando los diferentes módulos y algunos recursos lexicográficos disponibles

³¹ Figura obtenida de <http://www.fungramkb.com>, fecha de consulta 19/05/2014.

Si se observan los diferentes módulos lingüísticos, comenzando por el *nivel léxico*, se puede explicar sus componentes de la siguiente manera: el **Lexicón** es el módulo que contiene la información morfosintáctica, pragmática y colocacional de las unidades léxicas. Se trata, por tanto, de información que ayuda a utilizar dichas unidades de manera correcta formalmente en el discurso. El **Morficón** contiene aquellas reglas que afectan a la morfología flexiva, como por ejemplo las flexiones verbales o conjugaciones, o las concordancias de género y número. El siguiente módulo lingüístico, el *nivel gramatical*, contiene el Constructicón, que almacena los diferentes **Gramaticones** de las diferentes lenguas naturales, los cuales constituyen esquemas de construcciones que permiten a la RRG, teoría lingüística mencionada con anterioridad, construir la interfaz entre la sintaxis y la semántica, denominada *algoritmo de enlace*, para representar un texto de entrada en lenguaje natural mediante una estructura lógica. Asimismo, en FunGramKB esta estructura lógica se ve mejorada mediante un nuevo formalismo denominado “estructura lógica conceptual” (CLS por sus siglas en inglés, *Conceptual Logical Structure*), que maximiza la carga informativa y reduce la redundancia (Periñán Pascual y Arcas Túnez, 2010b; Mairal Usón *et al.*, 2012; Periñán Pascual y Mairal Usón, 2012; Van Valin y Mairal Usón, en prensa).

El otro gran módulo que compone FunGramKB es el *nivel conceptual*. Como se ha indicado anteriormente, el nivel conceptual almacena aquellas representaciones prototípicas de la realidad que el individuo recrea en su mente, no de las palabras con las que describe de manera lexicalizada dicha realidad. Se trata de un modelo basado en *esquemas*, entendido como los modelos de esquema enunciados por la psicología cognitiva (Periñán Pascual, 2012a), así como inspirado en las distinciones que Tulving establecía para el modelo de memoria a largo plazo (Periñán Pascual y Arcas Túnez, 2007a).

De este modo, mientras que el módulo de nivel léxico es particular para cada lengua natural (i.e. ha de crearse un Lexicón para el inglés, otro para el español y así sucesivamente para cada lengua con cada uno de los tres componentes del nivel léxico), el nivel conceptual es común a todos los lenguajes naturales, puesto que no está basado en conocimiento sobre las palabras sino en conocimiento sobre el mundo (Periñán Pascual y Arcas Túnez, 2006). Es por esto que se puede afirmar que el nivel conceptual de FunGramKB es universal, en tanto en cuanto los conceptos que recopila y categoriza son comunes al mundo: *“FunGramKB ontology takes the form of a universal concept taxonomy, where ‘universal’ means that every concept we can imagine has an appropriate place in this ontology.”* (Periñán Pascual y Arcas Túnez, 2007a:199). Se trata de la lexicalización de estos conceptos, la manera de organizarlos y de aglutinarlos en expresiones lingüísticas en las diferentes lenguas naturales lo que difiere entre ellas.

No obstante, el hecho de que el módulo conceptual de FunGramKB sea denominado *universal* no lo exime de ser lingüísticamente motivado, lo que no implica que sea lingüísticamente dependiente. Esta afirmación significa que cada uno de los conceptos introducidos en el módulo conceptual, particularmente en la Ontología, tiene necesariamente al menos una unidad léxica cuyo significado no coincide con ninguno de los postulados de significado ya presentes en la base de conocimiento. Asimismo, este proceso asegura que cada nueva unidad léxica que pueda surgir en el futuro, como consecuencia de la introducción de nuevas lenguas naturales en FunGramKB, tenga cabida dentro de ella tras un proceso de negociación. La posibilidad de introducción de estos nuevos conceptos, lexicalizados de una manera particular en una lengua natural, contribuye, además, a eliminar una posible carga subjetiva provocada por las propias lenguas maternas y contextos culturales de los ingenieros del conocimiento que realizan la tarea de poblar la base de conocimiento (Periñán Pascual y Arcas Túnez, 2010b).

El nivel conceptual de FunGramKB se compone de tres módulos (Periñán Pascual y Arcas Túnez, 2007a; Periñán Pascual y Arcas Túnez, 2010b):

- la **Ontología**, que es una estructura jerárquica de conceptos usados por las personas para describir cualquier situación cotidiana;
- el **Cognición**, que almacena conocimiento procedimental en forma de secuencias temporales o guiones, y que está basado en el modelo temporal de Allen (Allen, 1983; Allen y Ferguson, 1994);
- el **Onomástico**, que almacena conocimiento episódico acerca de entidades (p. ej., personas, ciudades, lugares, acontecimientos, etc.) en forma de bioestructuras (Periñán Pascual y Mairal Usón, 2010b).

Recordando lo mencionado previamente en este trabajo sobre las definiciones de ontología (cf. Capítulo II), se puede observar ahora que, en el ámbito de FunGramKB, *Ontología* se corresponde con uno de los módulos del nivel conceptual de la base de conocimiento: aquel módulo que recoge una jerarquía de los conceptos habituales que un hablante utiliza para expresar situaciones cotidianas. No obstante, no se ha de confundir esta serie de conceptos con las lexicalizaciones de los mismos. Esto es, los conceptos serían comunes a todas las lenguas naturales, pero la lexicalización de éstos es lo que variaría entre ellas. Sirva como ejemplo el concepto que en lenguaje natural podríamos definir como “*Mueble, por lo común de madera, que se compone de una o de varias tablas lisas sostenidas por uno o varios pies, y que sirve para comer, escribir, jugar u otros usos*”³². Una posible lexicalización en inglés podría ser *table* o *desk*; en español *mesa* (o también *escritorio*, por ejemplo), en alemán *Tisch* y así sucesivamente.

³² Definición de “mesa” por el Diccionario RAE, 22ª edición, accesible online: <http://lema.rae.es/drae/?val=mesa>, fecha de consulta 06/11/2013.

De este modo, *Ontología* en el marco de FunGramKB es uno de los elementos sobre los que se apoya la base de conocimiento, siendo así una parte más de dicha base. Este hecho enlaza con la idea mostrada previamente en alusión a la concepción de *ontología* que suele ser mantenida por cada autor, la cual habitualmente estará relacionada con el uso que éste haya hecho de las ontologías o con el modelo particular que utilice o que haya creado.

A continuación, se realizará una descripción más pormenorizada de los tres módulos del nivel conceptual de FunGramKB, puesto que este es el nivel en el que se centran las tareas mostradas en el presente trabajo y, más concretamente, en el módulo denominado Onomasticón.

La **Ontología** es el elemento central y eje fundamental de FunGramKB. Comprende tres tipos generales de conceptos, llamados metaconceptos y señalados con el símbolo #: entidades (# ENTITY), eventos (# EVENT) y cualidades (# QUALITY). Estos tres conceptos organizan la dimensión cognitiva de los nombres, verbos y adjetivos, respectivamente (Periñán Pascual y Arcas Túnez, 2007a). Estas tres dimensiones conceptuales se relacionan entre sí a través de los llamados postulados de significado (*Meaning Postulates*, MPs), que más adelante detallamos en profundidad. De manera adicional a las dimensiones de entidades, eventos y cualidades, actualmente se está trabajando en la posibilidad de crear una subontología de relaciones, lo cual correspondería lingüísticamente a las preposiciones, donde se ubicarían conceptos como, por ejemplo, +OVER_00 y otros de similar naturaleza: +ON_00, +UNDER_00, etc. Por tanto, en el presente trabajo, la aparición de dichos conceptos en las reglas

elaboradas para el mismo (Apéndice I) se debe entender como una inclusión, a priori, teórica³³.

El siguiente nivel bajo los metaconceptos se compone de los conceptos básicos, precedidos por el símbolo +, quedando por debajo de ellos los conceptos terminales, identificados mediante el símbolo \$. Todos ellos, tanto los conceptos básicos como los terminales, completan su nomenclatura mediante un sufijo numérico al final, al cual se antepone un guión bajo (p. ej., _00, _01, etc.). A modo de ilustración, podemos observar una figura que muestra la descripción de esta jerarquía de unidades conceptuales en Perrián Pascual y Arcas Túnez (2007a) para un ejemplo concreto, el concepto terminal \$FOOTBALL_00:

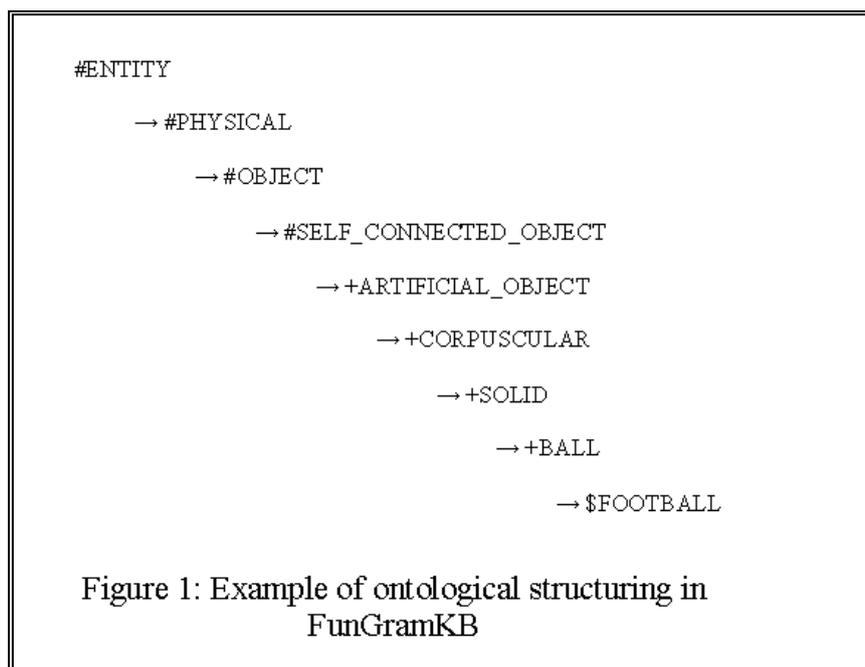


Figura 2. Ejemplo de estructuración ontológica en FunGramKB (Perrián Pascual y Arcas Túnez, 2007a:199).

El inventario de conceptos básicos de la Ontología procede, en su origen, del listado de vocabulario definitorio del diccionario *Longman Dictionary of Contemporary English* (Procter, 1978), denominado *Longman Defining Vocabulary*, el cual se ha

³³ J.C. Perrián Pascual, comunicación personal, 23 noviembre 2012.

sometido a un proceso de filtrado mediante la metodología COHERENT (*CO*nceptualization+ *Hi*Erarchization + *Re*modelling + *refineme*NT), que encontramos descrita en Perrián Pascual y Mairal Usón (2011). La siguiente figura resume el modo en que a través de la metodología COHERENT se lleva a cabo el tratamiento y depuración de los conceptos inicialmente listados en el inventario *Longman Defining Vocabulary*:

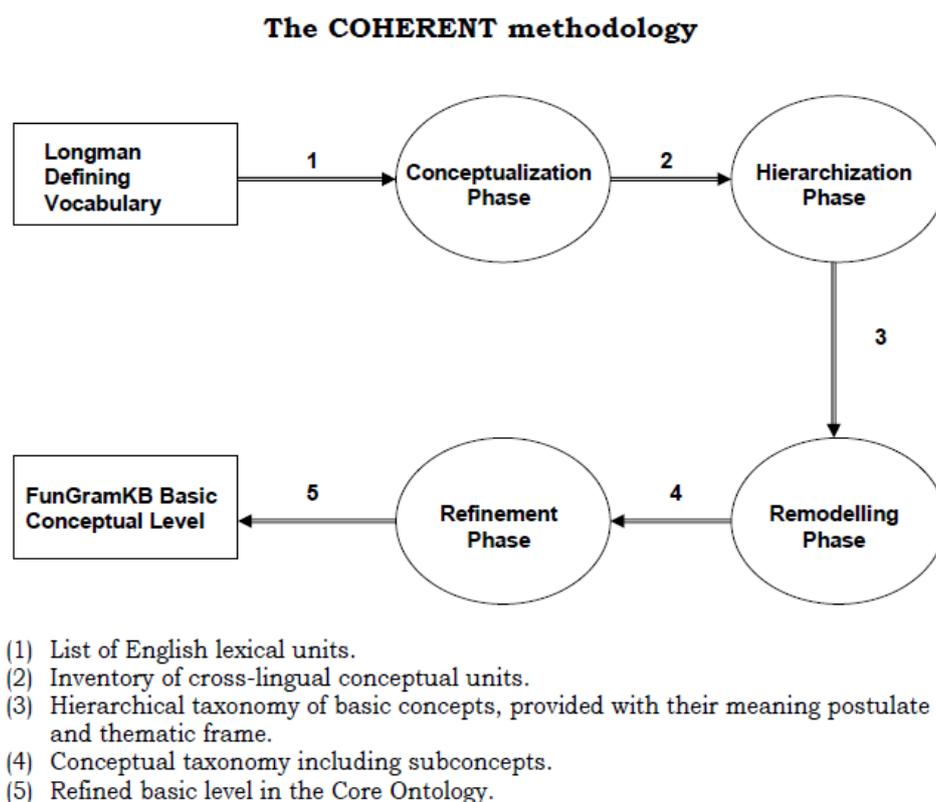


Figura 3. El procedimiento de la metodología COHERENT (Perrián Pascual y Mairal Usón, 2011:20).

Volviendo a la estructura de la Ontología de FunGramKB, ésta se compone de un módulo central (*core ontology*) y de otras sub-ontologías denominadas *satélites*, que abarcan terminologías específicas de dominios concretos. Por ejemplo, una ontología satélite de FunGramKB en la que se ha desarrollado un notable trabajo es la subontología de terminología legal y jurídica, que también incluye terminología relacionada con el terrorismo y el crimen organizado (Ureña Gómez-Moreno *et al.*,

2011; Felices Lago y Ureña Gómez-Moreno, 2012; Carrión Delgado, 2012; Perrián Pascual y Arcas Túnez, 2014).

El **Cognición** es el módulo que almacena conocimiento procedimental en forma de secuencias temporales o guiones de situaciones cotidianas (p. ej. ir al cine, visitar un museo, ir a un restaurante, etc.) y que está basado particularmente en el modelo temporal de Allen (Allen, 1983; Allen y Ferguson, 1994). En FunGramKB, se estructura como una serie de predicaciones dentro de un marco lineal temporal. Una predicación es una estructura conceptual que narra un evento mediante la interacción de sus participantes. Un guión, por tanto, comprende varias predicaciones y cada una de ellas se entiende como un evento E , tratado como un intervalo entre un par de puntos temporales: el punto temporal i , que es el inicio de la acción, y el punto temporal t , que es su fin. Así como Allen había previsto la posibilidad de diversas relaciones entre estos eventos (p. ej. simultaneidad, posterioridad o solapamiento, entre otros), también quedan éstas reflejadas en FunGramKB (Perrián Pascual y Arcas Túnez, 2010b).

Finalmente, el **Onomasticón** está formado por aquellas entidades nombradas (i.e. nombres propios) que componen nuestro conocimiento enciclopédico o cultural (p. ej. personas, obras artísticas, lugares, acontecimientos, organizaciones, eventos históricos, etc.), en forma de bio-estructuras. La naturaleza y características de este módulo son desarrolladas de manera más extensa en la sección 3.3 del presente capítulo, motivo por el cual a continuación trasladamos la discusión al lenguaje de representación utilizado en FunGramKB.

3.2. COREL COMO LENGUAJE DE REPRESENTACIÓN: LOS POSTULADOS DE SIGNIFICADO Y LOS MARCOS TEMÁTICOS

Como se ha mencionado anteriormente, los conceptos dentro de FunGramKB se relacionan a través de los llamados *postulados de significado*. La creación de estos constructos conceptuales responde a la necesidad de representar el conocimiento, sin perjuicio de que el Modelo Léxico Construccional siga, como modelo teórico escogido, impregnando los módulos lingüísticos de la base de conocimiento.

No obstante, además de la importancia que supone el contar con un modelo lingüístico teórico robusto tras el sistema, FunGramKB también ha de plantearse, como toda base de conocimiento, el lenguaje de representación mediante el cual se introducirán estos constructos conceptuales en el sistema. Este punto en la creación de la base supone una inflexión clave a la hora de perfilar el modelo, por lo que merece la pena reseñar las diferencias principales con otras bases de conocimiento y ontologías existentes.

Antes de poder comprender la elección de un lenguaje de representación u otro para la arquitectura de una base de conocimiento u ontología, es necesario efectuar un pequeño ejercicio de reflexión histórica. Como se ha señalado anteriormente, la conveniencia o ventaja de contar con una teoría lingüística que actúe de soporte para la creación de los módulos léxicos del modelo no siempre ha sido una realidad. Dada la naturaleza digital y computacional de estos recursos, las ontologías y las bases de conocimiento han sido tradicionalmente iniciadas, desde su etapa germinal, más por ingenieros de la computación y profesionales de la informática que por lingüistas o, al menos, pocas han sido fruto de la cooperación entre ambas disciplinas si miramos desde una perspectiva diacrónica. Esta necesidad, ahora obvia, de aunar diferentes perspectivas y áreas de conocimiento es algo que a día de hoy puede parecer natural,

pero que no siempre ha sido así (Mairal Usón y Perriñán Pascual, 2010a; Perriñán Pascual, 2012a).

Siendo, por tanto, en su mayoría ingenieros quienes han creado gran parte de los modelos y su arquitectura, y puesto que su disciplina de trabajo pertenece al campo de la computación, parece coherente que gran parte de los lenguajes de representación³⁴ escogidos pertenezcan al conjunto de los usados tradicionalmente en dicha disciplina, lo que actúa también de efecto colateral al hecho de contar con una base teórica más computacional que lingüística. Es así que la lógica de primer orden, por ejemplo, ha sido utilizada en numerosas ontologías y bases de conocimiento. Sin ir más lejos, CycL, el lenguaje formal utilizado por el proyecto Cyc mencionado en el Capítulo II, basa su sintaxis en el cálculo de predicados de primer orden, así como en KIF³⁵ (*Knowledge Interchange Format*). Otros lenguajes formales están basados también en la Semántica de Marcos (*Frame Semantics*) de Minsky (1974), como por ejemplo FLogic (Kifer y Lausen, 1989), siendo esta teoría combinada también con KIF para lenguajes como Ontolingua (Gruber, 1992), si bien posteriormente el propio Minsky (2000) recomendaba el uso de múltiples lenguajes para representar el conocimiento:

“There is no best way to represent knowledge. The present limitations of machine intelligence stem largely from seeking unified theories incapable of reasoning well, while our purely symbolic logical systems lack the uncertain, approximate linkages that can help us make new hypotheses. Human versatility must emerge from a

³⁴ No se ha de confundir lenguaje de representación con lenguajes de programación sobre los que se sostiene el sistema. Mientras que los primeros son los que el ingeniero del conocimiento utiliza para introducir constructos conceptuales relacionados con las unidades y los que, a su vez, el usuario utilizará en su interfaz con la base de conocimiento a la hora de recuperar la información que ésta pueda ofrecerle, los segundos serían los que sirven para implementar computacionalmente la herramienta (*caja negra*). Entendemos por *caja negra* cada módulo en el que se implementa el trabajo de una herramienta informática sin que sea necesario que seamos conscientes de todos y cada uno de los procesos que se llevan a cabo durante esa implementación, ya que lo que nos interesa es la entrada de elementos en el sistema y el resultado obtenido tras su procesamiento.

³⁵ KIF es un lenguaje computacional orientado al intercambio de conocimiento entre programas diferentes. Contiene semántica declarativa, comprensión lógica, metadatos sobre la representación del conocimiento y permite la representación de reglas de razonamiento monotónicas, haciendo posible la definición de objetos, funciones y relaciones.

large-scale architecture in which each of several different representations can help overcome the deficiencies of the other ones.”

(Minsky, 2000:71).

Incluso, el proyecto ConceptNet (Liu y Singh, 2004a; 2004b) da un paso más allá y postula la representación del conocimiento utilizando lenguaje natural, debido a la gran flexibilidad que éste ofrece frente a la lógica, visión ésta que podría ser rebatida por muchos. Si bien es cierto que la lógica parece conllevar, dentro de su rigor y exactitud, la desventaja anexa de ser demasiado estricta en algunos contextos, especialmente si no permite excepciones, el extremo de utilizar lenguaje natural para la representación del conocimiento podría fomentar una excesiva ambigüedad, lastre adherido a la cualidad de poder expresivo y flexibilidad. Perinián Pascual y Arcas Túnez (2007a) lo expresan de este modo:

“Dik (1997) proposes using words from the own language when describing meaning postulates, since meaning definition is an internal issue of the language. However, this strategy contributes to lexical ambiguity due to the polysemic nature of the defining lexical units. In addition, describing the meaning of words in terms of other words leads to some linguistic dependency (Vossen, 1994). Instead, FunGramKB employs concepts for the formal description of meaning postulates, resulting in an interlanguage representation of meaning.”

(Perinián Pascual y Arcas Túnez, 2007a:200).

Mediante esta mención a FunGramKB, retomamos a continuación la discusión y definición del lenguaje formal a través del cual se representan los constructos conceptuales en este modelo.

El lenguaje de representación del conocimiento utilizado en FunGramKB responde al nombre de **COREL** (*Conceptual Representation Language*)³⁶. Se trata de un lenguaje de interfaz para la representación formal del conocimiento conceptual. Esta característica hace que COREL sea, a la vez que un lenguaje formal robusto tratable por

³⁶ Para una completa descripción de la gramática del lenguaje COREL, referimos a Perinián Pascual y Mairal Usón (2010b).

la máquina, una interlengua fácil de interpretar por el usuario. La representación en COREL de los conceptos se realiza mediante el empleo de dos estructuras complementarias: los marcos temáticos y los postulados de significado (Periñán Pascual y Mairal Usón, 2009; 2010b).

Los **marcos temáticos** (*Thematic Frames*, TF, en FunGramKB) son estructuras conceptuales que explicitan el número y tipo de participantes que intervienen en la situación prototípica³⁷ retratada por cada concepto, lo que también incluye las preferencias de selección de los participantes (Periñán Pascual y Mairal Usón, 2009; 2010b). Gracias a los marcos temáticos, cada concepto encuentra delimitado el marco en el que la situación que describe tiene lugar. Para ilustrar esto, podemos tomar el ejemplo del concepto +PULL_00, el cual quedaría lexicalizado en inglés como *pull* o *draw* y en español con unidades léxicas como *tirar* o *estirar*:

(1)

(x1: +HUMAN_00 ^+ANIMAL_00)Agent (x2:+CORPUSCULAR_00)Theme
(x3)Location (x4)Origin (x5)Goal

La equivalencia, según explican Periñán Pascual y Mairal Usón (2009), en lenguaje natural de esta estructura sería la siguiente: “*Thus, the thematic frame (1) describes a prototypical cognitive scenario in which «entity₁ (Agent) moves entity₂ (Theme) from one place (Origin) to another (Goal), there also being a place (Location) along which entity₂ moves».*” (Periñán Pascual y Mairal Usón, 2009:267). Podemos añadir que, asimismo, según las preferencias de selección presentes en el marco temático, se sabe que la entidad₁ (el agente) debe ser o bien un humano, o bien un

³⁷ Recordemos que el MLC contempla teorías de la lingüística cognitiva, por lo que esta *prototipicidad* de la que se habla ha de entenderse en el sentido de la teoría de prototipos tratada, entre otros, por Lakoff y Langacker (Lakoff, 1987; Langacker, 1987; 1991) e iniciada por Rosch (1973).

animal (+HUMAN_00 ^ +ANIMAL_00) y que la entidad₂ (el tema) ha de ser una entidad sólida tridimensional (+CORPUSCULAR_00).

Por otro lado, una vez que se han detallado los participantes en la situación prototípica que cada concepto designa, se procede a crear el **postulado de significado** (*Meaning Postulate*, MP), que es la estructura que define a dicho concepto mediante el enfoque de la semántica profunda. Un postulado de significado es una secuencia de una o varias predicaciones, lógicamente conectadas, que son constructos conceptuales que evocan los rasgos genéricos del concepto (cf. Perinián Pascual y Arcas Túnez, 2004). Si continuamos con el ejemplo del concepto +PULL_00, observamos que Perinián Pascual y Mairal Usón (2009) expresan el postulado de significado de este modo:

(2)

+((e1: +MOVE_00 (x1)Agent (x2)Theme (x3)Location (x4)Origin (x5)Goal
(f1: +HAND_00 ^ +MOUTH_00)Instrument (f2: (e2: +SEIZE_00 (x1)Theme
(x2)Referent)) Condition) (e3:+BE_00 (x1)Theme (x5)Referent))

La estructura en (2), como se puede observar, incluye una co-indización de los participantes, denominados *argumentos*, señalados en el marco temático mostrado en (1), que podemos identificar mediante la letra *x* seguida de un número. Cada una de las predicaciones que enumeran una característica genérica del concepto está designada por la letra *e*, seguida de un número. En caso de que aparezca el signo + precediendo a una de estas predicaciones fuera del paréntesis, significará que dicha predicación es estricta, la cual permitirá exclusivamente la herencia monotónica. Por *monotónica* entendemos que no hay excepciones al contenido de dicha predicación, dentro de todas las situaciones, prototípicas o no, que puedan identificarse mediante el concepto que se está describiendo. Por ejemplo, si hablamos de un pájaro, una característica prototípica de este tipo de animales es “puede volar” pero, puesto que es posible que exista un pájaro

que, a pesar de serlo, no sea capaz de volar (p. ej. un pingüino o un avestruz), la característica “puede volar” será entonces una característica no estricta o *rebatible* del concepto “pájaro”, i.e., una característica que permite la herencia *no monotónica*.

Otra de las características de los postulados de significado es que pueden contener elementos llamados *satélites*, designados por la letra *f*, que aportan información periférica acerca de la situación conceptual descrita por las predicaciones. Dependiendo de la centralidad del papel en el evento que se está describiendo, a cada elemento participante se le asignará el papel de *argumento* o de *satélite* dentro del postulado de significado (Periñán Pascual y Mairal Usón, 2010b), por lo que se observa que, nuevamente, nociones como la centralidad hacen acto de presencia, en clara referencia a la teoría de prototipos anteriormente mencionada (cf. Nota 37). De este modo, la equivalencia del postulado de significado del ejemplo (2) en lenguaje natural sería la siguiente: “*A person or animal moves something towards themselves with their hand or mouth, providing that they hold it firmly.*” (Periñán Pascual y Mairal Usón, 2009:268). Por consiguiente, observamos que cualquier concepto puede ser representado en FunGramKB mediante el lenguaje COREL, lo que constituye un reflejo del compromiso de universalidad de esta base de conocimiento mencionado anteriormente (Periñán Pascual y Arcas Túnez, 2010a).

A continuación, procedemos a efectuar la descripción pormenorizada del módulo Onomasticón, ubicación central del trabajo desarrollado para la presente tesis doctoral.

3.3. EL ONOMASTICÓN DE FUNGRAMKB: ARQUITECTURA Y CONTENIDO

El Onomasticón de FunGramKB está encuadrado dentro del nivel conceptual de la base de conocimiento, junto a la Ontología y al Cognición. El Onomasticón, como se ha reseñado con anterioridad de manera breve, almacena en forma de bio-estructuras la información relativa a las instancias de entidades y eventos nombrados³⁸ que reflejan conocimiento cultural, entendido de la siguiente forma: “*Cultural knowledge consists of factual information about the past, present or future model of the world. This knowledge can be shared to a higher or lesser degree, depending on the individuals’ education [...].*” (Periñán Pascual, 2012b:195). Con la finalidad de comprender a qué nos referimos con la expresión *bio-estructura*, es necesario regresar al carácter prototípico del conocimiento almacenado en los módulos conceptuales de la Ontología y el Cognición.

Dada la prototipicidad de los elementos que pueblan la Ontología y el Cognición, llamamos *proto-estructuras* a los constructos cognitivos que los conforman, ya que se deben a la generalidad y no a la particularidad. No obstante, a pesar de reflejar esta generalidad, también puede suceder que en el marco de una situación prototípica encontremos excepciones a la regla. Esto significa que cuando cabe la posibilidad de que alguna de las características de la situación prototípica pueda, en algún momento hipotético del tiempo, ser diferente de las expresadas en la predicación, se utiliza la etiqueta de *rebatible*, lo que permitirá una herencia *no monotónica* y que se indica en lenguaje COREL mediante el empleo del signo * delante de la predicación en cuestión.

³⁸ Por *entidad nombrada* entendemos “[...] una palabra o secuencia de palabras que se identifican con un nombre de persona, organización o lugar.” (Periñán Pascual y Carrión Varela, 2011: 87), si bien esta denominación se puede extender a otro tipo de entidades tan diversas como hechos históricos o movimientos artísticos, entre otras.

Por el contrario, el Onomasticón refleja exactamente la situación opuesta, donde la prototipicidad deja paso a la especificidad. Puesto que se trata de entidades concretas existentes, se utiliza la etiqueta de *bio* para nombrar a estas estructuras. De este modo, mientras que una entidad de la Ontología es +SONG_00, una entidad del Onomasticón es %HEY_JUDE_00³⁹.

Además del parámetro de la prototipicidad, los esquemas conceptuales en FunGramKB también se clasifican conforme a otro parámetro: la temporalidad (Periñán Pascual y Arcas Túnez, 2010b). Teniendo en cuenta esta característica, el conocimiento dentro de los esquemas conceptuales puede presentarse de manera temporal o atemporal. Si se hace de manera temporal, significará que ese conocimiento se presenta dentro de un marco de tiempo determinado, al cual denominamos *macroestructura*. Por ejemplo, la biografía de una persona (i.e. una persona que se representa por medio de una unidad conceptual del Onomasticón) o un guión de los reflejados en el Cognición poseerían la propiedad de ser macroestructuras. Por otro lado, cabe la posibilidad de que las estructuras conceptuales representen el conocimiento de manera atemporal, lo que denominamos *microestructuras*, como sucedería con una característica aislada de una entidad del Onomasticón o un postulado de significado de la Ontología. La convergencia de estos dos parámetros, prototipicidad y temporalidad, resulta en la creación de una tipología de estructuras conceptuales compuesta por proto-microestructuras, proto-macroestructuras, bio-microestructuras y bio-macroestructuras. Si se coloca en una matriz esta combinación de parámetros, se obtendrá la siguiente tabla que lo ilustra:

³⁹ Nótese el símbolo % utilizado para identificar a las entidades del Onomasticón, diferente de los símbolos #, + o \$ que identifican a las entidades pertenecientes a la Ontología y que se corresponden con la clasificación previamente enumerada de metaconceptos, conceptos básicos y conceptos terminales, respectivamente.

		TEMPORALIDAD	
		—	+
P R O T O T I P I C I D A D	+	Proto-microestructura (Postulado de significado)	Proto-macroestructura (Guión)
	—	Bio-microestructura (Retrato)	Bio-macroestructura (Historia)

Tabla 1. Tipología de esquemas conceptuales en FunGramKB (Periñán Pascual y Arcas Túnez, 2010b:2668).

Además de estar organizadas de la manera arriba ilustrada, estas cuatro estructuras se integran en la base FunGramKB, lo que puede observarse representado de manera gráfica en la siguiente figura:

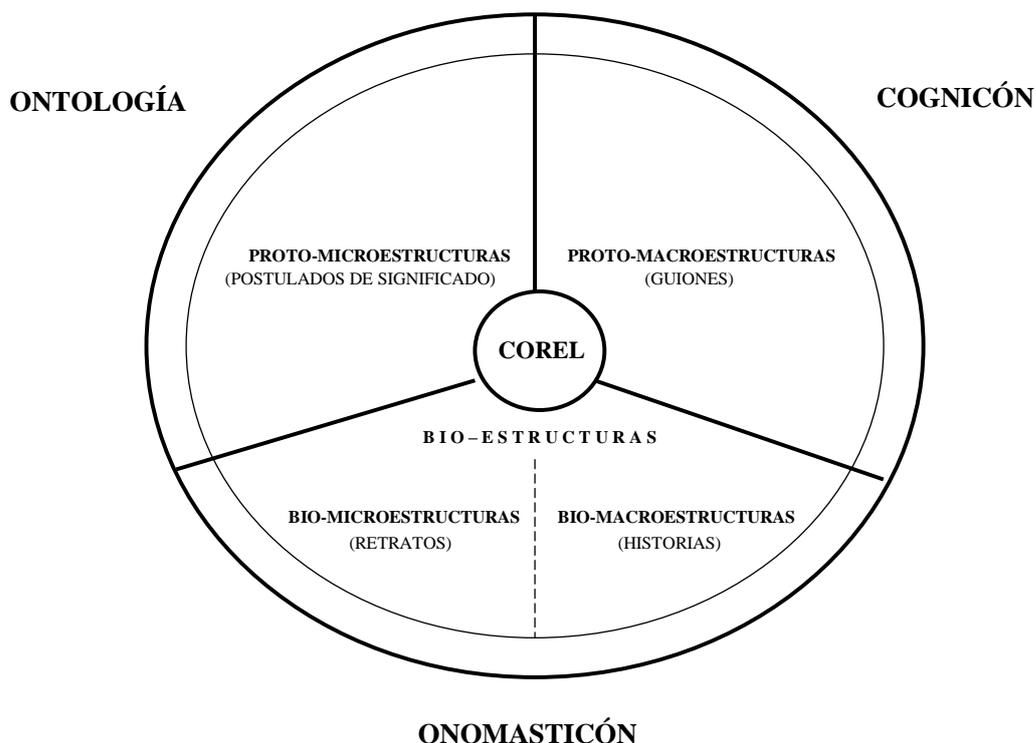


Figura 4. El Planeta Cognitivo (Periñán Pascual y Mairal Usón, 2010b).

Esta figura muestra la necesaria interrelación entre los tres módulos del nivel conceptual de FunGramKB a través de un mismo lenguaje de representación, COREL

(cf. sección 3.2), ya que, al igual que ocurre con los componentes de la memoria humana a largo plazo, un sistema de PLN que persiga permitir con éxito el razonamiento ha de prever que los componentes que lo forman puedan relacionarse entre ellos a través de un mismo lenguaje (Periñán Pascual y Arcas Túnez, 2010b; Periñán Pascual y Mairal Usón, 2010b).

La Tabla 1 y la Figura 4 arriba mostradas sirven de contexto clave a la hora de describir el Onomasticón y los elementos que éste alberga. De este modo, si se toma una entidad como, por ejemplo, *Elvis Presley* (%ELVIS_PRESLEY_00 en notación COREL), dependiendo del tipo de característica que se desee reflejar se necesitará un tipo u otro de estructura conceptual. Por tanto, si se desea indicar que la profesión de Elvis es cantante, será preciso emplear una bio-microestructura, ya que dicha propiedad puede encuadrarse como uno de los muchos rasgos que, de manera relativamente aislada del resto de características, describen a Elvis en un momento concreto del tiempo, como serían, entre otras muchas, la profesión de actor, su color de pelo o su estatura (cf. Periñán Pascual y Arcas Túnez, 2010b). Es por ello que este tipo de microestructuras en el Onomasticón se denominan “retratos” (*snapshots*) por el hecho de asemejarse a una fotografía estática que se toma en un momento particular. Sin embargo, si se desea describir la biografía de Elvis, es necesario hacer uso de una bio-macroestructura, las llamadas “historias” (*stories*) por obvias razones: contienen elementos que han de ser integrados dentro de un esquema temporal determinado. En el ejemplo que nos ocupa, se podrían enumerar como historias, entre otras, la fecha de nacimiento o de fallecimiento del cantante, ya que se trata de elementos cuya existencia se encuadra en un marco temporal establecido y cuyo orden no es posible alterar.

Otro ejemplo al que se puede aludir es al de una entidad como el Taj Majal. De nuevo, si se persigue describir una propiedad como “hecho de mármol” o “mausoleo”,

se haría mediante una bio-microestructura (retrato), mientras que el relato de su proceso de construcción habría de realizarse mediante una bio-macroestructura (historia). Esto se puede ilustrar en lenguaje COREL a través del ejemplo (3), originalmente mostrado en Perinián Pascual y Arcas Túnez (2010b) acerca de la entidad %TAH_MAHAL_00. El equivalente en lenguaje natural de las predicaciones *e1*, *e2* y *e3* representadas en (3-a) se muestra en (3-b):

(3-a)

+*(e1: +BE_02 (x1: %TAH_MAHAL_00)Theme (x2: %INDIA_00)Location)*
**(e2: +BE_01 (x1)Theme (x3: +WHITE_00 & \$MARBLE_00)Attribute)*
**(e3: +COMPRISE_00 (x1)Theme (x4: 1 \$DOME_00 & 4+TOWER_00)Referent)*

(3-b)

El Taj Majal está ubicado en India.

Su principal material es el mármol blanco.

El Taj Majal tiene una cúpula principal y cuatro torres.

Estas tres predicaciones acerca del Taj Majal pertenecen a la categoría “retrato”, ya que las tres enumeran rasgos atemporales de la entidad. Sin embargo, apreciamos que sólo la primera de ellas porta el símbolo + al frente, lo que indica que tan sólo esa predicación es estricta. Las otras dos predicaciones son rebatibles, ya que si el Taj Majal encontrara destruida una de sus torres o la cúpula principal por cualquier circunstancia, o si alguna de éstas se sustituyera por un elemento hecho de otro material diferente del mármol blanco, el Taj Majal seguiría considerándose la misma entidad. No obstante, si el Taj Majal no se encontrara en la India, entonces ya no se trataría de la misma entidad, sino de una reproducción del mausoleo en otra ubicación.

Por otro lado, y continuando con el mismo ejemplo, también es posible representar características temporales del Taj Majal. Es el caso de las siguientes

predicaciones *e1* y *e2* mostradas en (4-a) y su equivalente en lenguaje natural en (4-b) (Periñán Pascual y Arcas Túnez, 2010b):

(4-a)

+*(e1: past +BUILD_00 (x1)Theme (x2: %TAH_MAHAL_00)Referent (f1: 1633)Time)*
+*(e2: past +BE_00 (x2)Theme (x3:% WORLD_HERITAGE_SITE_00)Referent (f2: 1983)Time)*

(4-b)

El Taj Majal fue construido en 1633.

El Taj Majal se convirtió en Patrimonio de la Humanidad por la UNESCO en 1983.

Los conceptos representados en (4) constituyen *historias* ya que, además de ser predicaciones estrictas, lo que está indicado por el signo + frente a cada una de ellas⁴⁰, pertenecen a un esquema temporal en el que están encuadradas de una manera precisa y determinada. Por consiguiente, se observa a través de los ejemplos anteriormente mencionados cómo una misma entidad puede compartir retratos e historias que representen los rasgos conceptuales que la definen.

Mediante la comparación de los dos tipos de estructuras del Onomasticón (i.e. retratos e historias) con los elementos de los otros dos módulos conceptuales de FunGramKB, la Ontología y el Cognición, es comprensible que los postulados de significado de la Ontología se consideren proto-microestructuras, por su carácter estático y atemporal, mientras que los guiones del Cognición son proto-macroestructuras, ya que los eventos dentro de ellos siguen un esquema temporal determinado.

⁴⁰ No es posible cambiar el año de construcción del Taj Majal, ni aquel en el que fue declarado Patrimonio de la Humanidad.

No obstante, y paralelamente a las similitudes entre módulos conceptuales que han sido enumeradas hasta ahora, uno de los principales rasgos distintivos del Onomasticón frente a la Ontología y al Cognición es que la manera de poblar este módulo no se realiza de manera enteramente manual o asistida, como sucede en los otros dos módulos, sino que se realiza de manera semi-automática (Periñán Pascual y Arcas Túnez, 2010b). Precisamente, el presente trabajo tiene como primer objetivo el desarrollo de una metodología que permita poblar de manera semi-automática el Onomasticón de FunGramKB, metodología que se ha esbozado inicialmente en Carrión Varela (2010) y Periñán Pascual y Carrión Varela (2011), la cual detallamos en profundidad en el Capítulo IV. Antes de definir esta metodología, es pertinente en este momento hacer una mención a los sistemas y estructuras pertenecientes a otros proyectos que podrían semejarse al Onomasticón de FunGramKB y, lo que es más importante, cuáles de ellos van a desempeñar un papel relevante como recursos para la población semi-automática de nuestra base de conocimiento.

3.4. ESTRUCTURAS RELACIONADAS: YAGO, YAGO2 Y DBPEDIA

Dentro del vasto universo ontológico mencionado en el Capítulo II del presente trabajo, podemos destacar tres proyectos concretos que se relacionan de manera especial con el Onomasticón de FunGramKB, en tanto en cuanto todos ellos versan sobre entidades nombradas y la manera de categorizarlas: el proyecto YAGO, su continuación YAGO2 y el proyecto DBpedia. Una importante razón por la cual se han escogido estos tres proyectos en particular, en lugar de otros similares, responde a que todos ellos toman la información acerca de las entidades nombradas de fuentes que son ricas en datos y, lo que es primordial, que se hallan en permanente actualización. Esto permite que en dichas fuentes tengan cabida nuevas entidades que puedan crearse, así como que la información relativa a las entidades existentes pueda ser actualizada a medida que

ésta cambie. Una de las más profusas fuentes de información que en la actualidad almacena conocimiento sobre millones de entidades es el proyecto Wikipedia.

Wikipedia⁴¹, un proyecto de la Fundación Wikimedia, es una enciclopedia multilingüe que contiene más de 15 millones de artículos en 285 lenguas naturales, libremente accesible en la web y editada de manera cooperativa por voluntarios de todo el mundo, bajo la supervisión de un grupo de administradores de cada edición que velan por el mantenimiento de los artículos y el cumplimiento de las normas. Este conocimiento, no obstante, a pesar de su dimensión y organización jerárquica, carece de estructuración conceptual, por lo que parece razonable pensar que reutilizarlo para dotarlo de este tipo de contenido e integrarlo en una base de conocimiento es una tarea que merece la pena acometer. Partiendo de esta idea, a continuación abordamos la descripción de los proyectos YAGO, YAGO2 y DBpedia, los cuales también reconocen desde el momento de su creación el gran valor informativo que Wikipedia puede ofrecer, así como la utilidad de organizarlo dentro de una ontología o una base de conocimiento. Si bien son numerosos los proyectos emprendidos a nivel global con el fin de reutilizar la información contenida en Wikipedia, nos centramos solamente en los tres mencionados con anterioridad debido a su proximidad con el modelo de FunGramKB, así como por la posibilidad de reutilización de los mismos que FunGramKB puede contemplar. Asimismo, hacemos mención también al trabajo de Bhole *et al.* (2007) y al sistema PYTHIA de Unger y Cimiano (2011) en relación a su conexión con YAGO2 y con DBpedia, respectivamente.

⁴¹ <http://www.wikipedia.org>

3.4.1. YAGO

El proyecto YAGO⁴² (Suchanek *et al.*, 2007, 2008; De Melo *et al.*, 2008; Kasneci *et al.*, 2008) es una ontología de entidades y relaciones que contiene más de un millón de entidades y más de 5 millones de relaciones. Se trata de un proyecto que pretende unificar información extraída de manera automática de Wikipedia y enriquecerla con la base de datos léxica WordNet⁴³ (Miller, 1995; Fellbaum, 1998) de manera que esta última se vea mejorada en tamaño y en contenido mediante la inclusión de información relativa a entidades nombradas, tales como personas, lugares, etc. Las relaciones en YAGO se explicitan mediante la estructura Is-A, así como a través de otras relaciones no taxonómicas como, por ejemplo, la relación de transitividad, la de aciclicidad u otras relaciones semánticas específicas como “HASWONPRIZE”, “LOCATEDIN”, “BORNINYEAR”, etc., dependiendo del contenido semántico específico que se desee reflejar. A continuación, presentamos algunos ejemplos que ilustran la manera en la que YAGO codifica la información (Suchanek *et al.*, 2007:698-9):

(5)

- a) ”Einstein” MEANS AlbertEinstein
- b) AlbertEinstein TYPE physicist
- c) physicist SUBCLASSOF scientist
- d) subclassOf TYPE transitiveRelation

En (5-a) se observa una redirección desde el texto que el usuario introduce en el momento de hacer la consulta, “Einstein”, hacia la entidad almacenada en YAGO, *AlbertEinstein*⁴⁴. Puesto que las palabras también se consideran entidades⁴⁵, es por ello

⁴² YAGO es el acrónimo de *Yet Another Great Ontology*.

⁴³ <http://wordnet.princeton.edu>

⁴⁴ Los autores reconocen la posibilidad de que una misma palabra, en este caso “Einstein”, se refiera a más de una entidad, como por ejemplo el músico Alfred Einstein, ejemplo mencionado en Suchanek *et al.* (2007), de lo que se deriva que en algún punto del proceso sería preciso efectuar una desambiguación.

⁴⁵ “Einstein” se encuentra entrecomillado en YAGO, como muestra el ejemplo (5-a), para expresar que se trata de una palabra.

que YAGO permite esta redirección mediante la relación MEANS. En los ejemplos (5-b) y (5-c) se reflejan dos de las múltiples propiedades de la entidad *AlbertEinstein* que YAGO almacena: en primer lugar, la entidad *AlbertEinstein* es una instanciación de la clase *physicist*, una correspondencia que se expresa mediante la relación denominada TYPE, mientras que, a su vez, la clase *physicist* está encuadrada bajo la clase más numerosa *scientist*, expresado mediante la relación denominada SUBCLASSOF. Finalmente, en (5-d) se observa que las relaciones también se consideran entidades, por lo que la relación del tipo *subclassOf* arriba mostrada, en minúscula en esta ocasión por ser considerada entidad, es una instanciación de la clase *transitiveRelation*.

Si comparamos YAGO con el Onomasticón de FunGramKB, encontramos diferencias notorias: en primer lugar, YAGO es una base de datos de conocimiento relacional, no de semántica profunda, lo que desde el inicio sitúa a FunGramKB en una posición frontal a ella. Incluso la base de datos WordNet, de la cual YAGO extrae la información, consiste en una red léxica, no en una base de conocimiento asentada sobre análisis de semántica profunda, lo que nos devuelve al planteamiento ya mencionado acerca de las bases relacionales.

A pesar de que una base de conocimiento relacional puede presentar ventajas a la hora de representar el conocimiento, se puede objetar que un sistema que define en relación a otros términos corre el riesgo de caer en la circularidad definitoria, así como en problemas derivados de la polisemia de las unidades léxicas, lo que en última instancia deriva en una robustez reducida a la hora de inferir información, disminuyendo, de esta forma, la eficiencia global de la herramienta. Obviamente, esto ocurre debido al empleo de unidades léxicas en lugar de conceptos: un concepto es unívoco en sí, a pesar de que pueda evocar más de una unidad léxica, pero una unidad léxica siempre puede tener la posibilidad de ser polisémica y evocar varios conceptos.

Los propios creadores de YAGO transmiten algunos casos de necesidad de desambiguación en procesos automáticos, los cuales han de corregir de manera manual⁴⁶. Consecuentemente, y a pesar de reconocer los esfuerzos realizados por la ontología YAGO, finalmente no se ha estimado que ésta resulte adecuada como recurso para la población semi-automática del Onomasticón de FunGramKB. No obstante, a continuación presentaremos las posibilidades de vinculación con FunGramKB que la mejora de YAGO, denominada YAGO2, puede aportar.

3.4.2. YAGO2

La herramienta YAGO ha experimentado desarrollos posteriores a su concepción inicial, los cuales han dado como fruto una versión mejorada y enriquecida denominada YAGO2 (Hoffart *et al.*, 2010). Se trata de una extensión de la base de conocimiento YAGO, cuya principal mejora es que tanto las entidades como los hechos y eventos contenidos en ella han sido encuadrados en una dimensión espacio-temporal. Para ello, YAGO2 toma datos de manera automática de tres fuentes principales, que son Wikipedia, GeoNames y WordNet (Hoffart *et al.*, 2010).

De este modo, en YAGO2 a las entidades, por ejemplo, se les asigna un rango temporal de existencia, mientras que los hechos pueden o bien tener un punto concreto de ocurrencia si se consideran instantáneos, o tener un rango temporal cuando poseen una fecha de inicio y de fin conocidas. Este fenómeno (i.e. la ubicación espacio-temporal de las entidades y los hechos) puede entrañar cierto paralelismo con algunos aspectos de FunGramKB, si bien con matices. Como hemos visto en la descripción del módulo Onomasticón, a la hora de conferir una dimensión temporal a las entidades y hechos, en FunGramKB encontramos una taxonomía dual (i.e. retratos e historias) que

⁴⁶ Un ejemplo de este fenómeno se cita en Suchanek *et al.* (2007), donde se menciona la palabra “*capital*”, entendida principalmente como “*capital city*” en Wikipedia y confundida frecuentemente con “*financial assets*”, el cual resulta ser el sentido más frecuente en WordNet.

se aplica a cada una de las propiedades dentro del catálogo vinculado a cada clase. Recordemos que los retratos son propiedades instantáneas tomadas en un momento particular, mientras que las historias contienen elementos que han de ser integrados dentro de un esquema temporal determinado (p. ej. fechas, procesos, logros, etc.). Así pues, cada una de las propiedades que definen una clase se debe encuadrar en una de estas dos tipologías: o bien es una propiedad que refleja una característica instantánea, o bien se trata de un rasgo anclado en un momento concreto de un proceso temporal. De esta forma, la dimensión temporal está siempre presente en el Onomasticón. Por tanto, observamos que, al contrario de lo que sucede en YAGO2, en FunGramKB son las propiedades las que definen a las entidades y las que poseen una naturaleza temporal determinada (retratos o historias), no siendo las propias entidades las que se pueden considerar o bien instantáneas o bien organizadas dentro de un rango temporal. Con objeto de clarificar algo más este fenómeno, observemos a continuación la clasificación que se efectúa en YAGO2 entre *Entidades* y *Hechos*:

“• Entities are assigned a time span to denote their existence in time. For example, Elvis Presley is associated with 1935-01-08 as his birthdate and 1977-08-16 as his time of death. Bob Dylan (who is still alive), is associated only with the time of birth, 1941-05-24. [...]

• Facts are assigned a time point if they are instantaneous events, or a time span if they have an extended duration with known begin and end. For example, the fact BobDylan created BlondeOnBlonde is associated with time point 1966-05-16 (the release date of this album). The fact BobDylan isMarriedTo SaraLowndes is associated with the time span from 1965-##-## to 1977-##-##. [...]

(Hoffart et al., 2010:12)

Frente a esto, en FunGramKB la entidad %BOB_DYLAN_00 (perteneciente a la clase “Person”) posee una amplia serie de propiedades entre las que se incluyen, por ejemplo, *birthDate*, *birthName* o *influencedBy*. En FunGramKB es cada propiedad la que recibe la calificación de historia o retrato, no la entidad en sí. En este caso, la

propiedad *birthDate* se consideraría historia, ya que se enmarca dentro de un rango temporal determinado, mientras que *influencedBy* sería un retrato, puesto que las influencias tomadas por una persona, tanto artísticas como de otra naturaleza, pueden variar en el tiempo y no estar necesariamente encuadradas dentro de un marco temporal fijo. Por tanto, no es la entidad %BOB_DYLAN_00 la que se enmarca dentro de una secuencia temporal, ni la que se considera instantánea o no, sino que son las propiedades que la definen las que deben ser clasificadas conforme a dicha taxonomía.

No obstante, se ha de puntualizar que quizás los valores temporales de YAGO2 vayan más allá de la mera etiqueta de características de una entidad o anotaciones de hechos (cf. Hoffart *et al.*, 2010:43), para ser puntos de anclaje en el espacio y en el tiempo. Este anclaje espacio-temporal es un fenómeno que, en cierto modo, ya está cubierto por las propiedades de cada clase del Onomasticón de FunGramKB, siempre que éstas denoten un valor temporal o espacial concreto de existencia. Por ejemplo, si una entidad en FunGramKB posee las propiedades *buildingStartDate*, *buildingEndDate*, *openingDate* y *address* (lo que sucede, por ejemplo, con las entidades adscritas a la clase “Building”), la suma de estas propiedades y los valores que obtengan nos darán automáticamente puntos de anclaje de dicha entidad en el tiempo y el espacio. Por tanto, podemos concluir que la inclusión de dimensiones temporales tanto en YAGO2 como en FunGramKB diverge, principalmente, en que tratan de reflejar diferentes tipos de naturaleza temporal.

Sin embargo, y a pesar de los patentes contrastes entre ambos enfoques, podría ser factible considerar que los puntos de vista de YAGO2 y del Onomasticón de FunGramKB puedan ser paralelos o, incluso, complementarios en determinados casos (p. ej., añadiendo anclajes de tiempo y espacio siguiendo la metodología de YAGO2 para aquellas entidades del Onomasticón cuyas propiedades definitorias no aportan

datos espacio-temporales), lo que podría sugerir la realización de un estudio de compatibilidad para la integración de ambos enfoques en FunGramKB en un futuro. Esta posibilidad supondría un paso más en la reusabilidad e interoperabilidad de las herramientas y modelos ontológicos, algo que YAGO ya realiza mediante su aportación de datos a DBpedia o su integración en la ontología SUMO⁴⁷ (De Melo *et al.*, 2008).

Esta última acción, la integración de YAGO en SUMO, resulta interesante para efectuar un paralelismo comparativo con FunGramKB, puesto que lo que se pretende en la fusión de YAGO y SUMO es interconectar una gran ontología formal axiomática (SUMO) que sea capaz de aportar algo de “sentido común”, con una base de conocimiento de construcción automática (YAGO) que contenga información factual sobre un gran volumen de hechos y entidades (cf. De Melo *et al.*, 2008). La intención de esta fusión podría considerarse análoga – si bien evidentemente menos cohesionada – a la interconexión existente entre los diferentes módulos que componen FunGramKB y, en concreto, entre los módulos conceptuales Ontología, que contiene primordialmente conocimiento conceptual del mundo y Onomasticón, el cual está principalmente poblado con conocimiento enciclopédico acerca de entidades nombradas. Tanto un lenguaje común para todo FunGramKB (i.e. el lenguaje COREL) como un planteamiento integrador de conjunto desde un principio hacen que la integración de YAGO y SUMO no pueda compararse con la completa y planeada interconexión entre los módulos de FunGramKB.

Regresando a YAGO2 y con respecto a la informatividad y contenido de la herramienta, ésta mantiene un enfoque conservador puesto que, en caso de duda, aboga por dejar sin asignar valor temporal a algunas entidades, a pesar de que sean temporalmente dependientes, en lugar de establecer un tiempo no definido

⁴⁷ SUMO es el acrónimo de **S**uggested **U**pper **M**erged **O**ntology.

correctamente. En este sentido, YAGO2 comparte con FunGramKB la filosofía de no asignar contenido (semántico, temporal, etc.) del que no se tenga certeza de su corrección, incluso a riesgo de provocar un vacío conceptual. De esta forma, se puede afirmar que quizás no esté todo, pero todo lo que está es cierto.

En relación a la dimensión espacial de las entidades, segunda característica que YAGO2 aporta como mejora de YAGO, YAGO2 obtiene de Wikipedia y de GeoNames⁴⁸ las coordenadas geográficas de ubicación de los lugares, así como otros datos de Wikipedia, al igual que ocurre en el Onomasticón de FunGramKB. Sin embargo, en comparación con FunGramKB, ésta no determina actualmente que las coordenadas de ubicación geográfica de una entidad resulten de un valor suficiente para el PLN como para reflejar dicha propiedad en el Onomasticón mediante la creación de una regla *ad hoc*, lo que presenta una diferencia con YAGO2 acerca de la relevancia que se le otorga a esta característica.

Un paralelismo existente entre YAGO2 y FunGramKB consiste en el hecho de que ambas bases de conocimiento poseen elementos denominados *reglas* que contribuyen a inferir información mediante los mecanismos de razonamiento que cada base ejecuta. No obstante, estas reglas difieren entre ambas bases tanto en la forma como en la noción sobre la que se edifican. En FunGramKB, como vemos con detalle en el Capítulo IV, lo que llamamos *reglas* son plantillas en lenguaje COREL que, a modo de predicaciones de un postulado de significado, son elaboradas de manera asistida por el ingeniero del conocimiento. Una vez elaboradas e introducidas en el Onomasticón, estas plantillas permiten que se efectúe la proyección automática del conocimiento desde otras fuentes (en concreto, DBpedia) a FunGramKB. De este modo, las plantillas COREL del Onomasticón se comportan como herramientas que posibilitan

⁴⁸ <http://www.geonames.org>

la proyección del conocimiento, actuando como vehículos transportadores de la información.

Sin embargo, en YAGO2 estas reglas responden tanto a necesidades de la propia arquitectura de la base de conocimiento como a la necesidad de rediseñar el código de extracción, con objeto de permitir una fácil extensión de las reglas que inicialmente se utilizaban en YAGO (Hoffart *et al.*, 2010). Existen diferentes tipos de reglas que posibilitan diferentes funciones. Por ejemplo, encontramos las denominadas *Reglas Factuales*, que son traducciones en forma declarativa de todas las excepciones y hechos que el previo código de YAGO contenía y que eran definidas de manera manual (Hoffart *et al.*, 2010). Un ejemplo de Regla Factual sería aquella que se utiliza para desambiguar unidades léxicas polisémicas. Para ello, se crea una relación denominada “*hasPreferredMeaning*” y se conecta a una unidad léxica con el código de WordNet vinculado al significado más usual de la misma. Por ejemplo, para indicar que la unidad léxica “*capital*” se refiere de manera principal a una ciudad y no a bienes financieros, siendo éste último el significado más común en WordNet, se indicaría mediante la siguiente regla: ““*capital*” *hasPreferredMeaning* *wordnet_capital_108518505*” (Hoffart *et al.*, 2010:10). Otras reglas existentes en YAGO2 son las *Reglas Implicativas*, que son reglas lógicas que sirven para deducir nuevo conocimiento a partir del conocimiento existente. De este modo, una Regla Implicativa enunciaría que si una relación es una sub-propiedad de otra relación, entonces todas las instancias de la primera relación son también instancias de la segunda (Hoffart *et al.*, 2010). Un tercer tipo de reglas son las *Reglas de Sustitución*, que permiten “limpiar” la base YAGO2 y normalizar formatos, como por ejemplo: ““*\{USA\}*” replace “*[[United States]]*” (Hoffart *et al.*, 2010:10). Finalmente, encontramos las *Reglas de Extracción*, que son capaces de generar automáticamente una secuencia de hechos si una parte del texto fuente coincide con una

expresión concreta (Hoffart *et al.*, 2010), de forma que se puedan identificar y extraer de mejor manera las entidades de una secuencia de texto. Estas últimas reglas son útiles para vigilar que no se extraiga información de las fuentes (por ejemplo, de Wikipedia) que sea sintácticamente errónea.

A la luz de esta comparación, podría decirse que la producción de reglas dentro de FunGramKB se consideraría más eficiente, puesto que no hay un enfoque concreto en dos dimensiones específicas – el tiempo y el espacio – sino que se muestra todo un catálogo de propiedades vinculadas a cada clase, aplicables a cada una de las entidades que contiene. Quizás en el futuro los creadores de YAGO2 intenten acometer la inclusión de más dimensiones cognitivas dentro de la base de conocimiento en adición al tiempo y el espacio, como pueden ser participantes, objetos, etc., para lo cual parece ser necesario crear un completo proceso de algoritmos para cada dimensión (i.e. un algoritmo para la inclusión de participantes, otro para la inclusión de objetos y así sucesivamente para cada dimensión). En contraposición a esto, tal y como demostramos en el siguiente capítulo, FunGramKB es capaz de reflejar de manera más eficiente estas dimensiones cognitivas, de muy diversa naturaleza, a través de las reglas creadas e introducidas en el Onomasticón.

Como cierre a la descripción de YAGO2, podemos mencionar un proyecto con el que dicho modelo parece presentar ciertas similitudes, el cual encontramos descrito en el trabajo de Bhole *et al.* (2007). El proyecto desarrollado por estos autores tiene como objetivo presentar las interconexiones existentes dentro de un marco temporal entre las entidades nombradas de Wikipedia, lo que resumen de este modo: “*Our goal is to generate a dynamic graph with named entities and relations between them, such as; a person was born at date X at city of Y, in year Z he/she lived in city W*”. (Bhole *et al.*, 2007:464).

Para ello, en primer lugar se efectúa una detección de las páginas o artículos más relevantes de las categorías “Place”, “Person” y “Organisation”, las cuales resultan ser las más populares de Wikipedia (cf. Tabla 7) para, a continuación, llevar a cabo la extracción de los datos sobre las entidades pertenecientes a dichas categorías. De este modo, el objetivo final se cumple mediante la presentación de líneas temporales en las que coexisten diferentes entidades nombradas (p. ej. personas, lugares, organizaciones, etc.), con el fin de mostrar las posibles interconexiones entre ellas. Entendemos que, de algún modo, YAGO2 parece recoger el testigo de esta tarea de conferir una ubicación espacio-temporal a las entidades nombradas, si bien en Bhole *et al.* (2007) se presenta de manera gráfica a través de diagramas que muestran las interconexiones entre entidades. Como ejemplo de este resultado, concluimos esta sección mostrando uno de los diagramas obtenidos tras la aplicación del método desarrollado por Bhole *et al.* (2007):

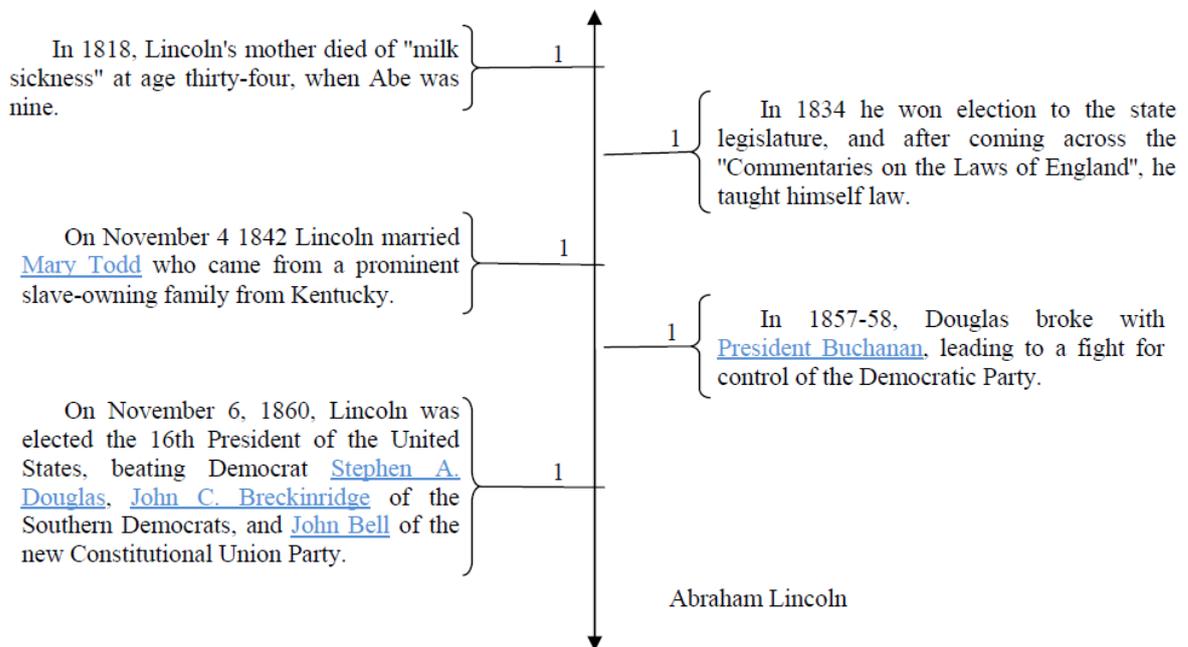


Figura 5. Ilustración de la línea temporal generada para Abraham Lincoln mostrando su conexión con otras personas identificadas (Bhole *et al.*, 2007:467).

3.4.3. DBPEDIA

En la presente sección mencionaremos un proyecto de alta relevancia para el Onomasticón de FunGramKB, denominado DBpedia⁴⁹ (Auer *et al.*, 2007; Bizer *et al.*, 2009). Si bien DBpedia tiene en común con los proyectos anteriormente mencionados el hecho de que obtiene datos de manera automática de Wikipedia, el resultado y la presentación de esta información difieren significativamente.

El proyecto DBpedia persigue extraer información estructurada procedente de Wikipedia para hacerla accesible en la web a través de la llamada *Web of Data*, después de haberla transformado en una rica base de conocimiento. La principal fuente de información para DBpedia dentro de cada artículo de Wikipedia son las fichas informativas (*info-boxes*), algo que constituye una práctica habitual en proyectos similares⁵⁰. Una ficha informativa, situada generalmente a la derecha en la parte superior de un artículo, contiene resumida la información más relevante que se puede encontrar dentro del mismo, de manera estructurada en forma de una relación Is-A (p. ej., para una persona figurará: lugar de nacimiento, año de nacimiento; para un lugar, su ubicación, extensión, etc.). La ilustración a continuación muestra una de estas fichas informativas:

⁴⁹ <http://dbpedia.org>

⁵⁰ Por ejemplo, véase el trabajo de Bhole *et al.* (2007) o el sistema PHYTHIA, de Unger y Cimiano (2011).

```
{{Infobox Town AT |
name = Innsbruck |
image_coa = InnsbruckWappen.png |
image_map = Karte-tirol-I.png |
state = [[Tyrol]] |
regbzkg = [[Statutory city]] |
population = 117,342 |
population_as_of = 2006 |
pop_dens = 1,119 |
area = 104.91 |
elevation = 574 |
lat_deg = 47 |
lat_min = 16 |
lat_hem = N |
lon_deg = 11 |
lon_min = 23 |
lon_hem = E |
postal_code = 6010-6080 |
area_code = 0512 |
licence = I |
mayor = Hilde Zach |
website = [http://innsbruck.at] |
}}
```



Ilustración 2. Ficha informativa (*Info-box*) de Wikipedia (Periñán Pascual y Carrión Varela, 2011:93).

Actualmente, la versión en lengua inglesa de DBpedia contiene más de 4 millones de entidades, de las cuales 3,22 millones están clasificadas en una ontología jerarquizada, incluyendo personas, lugares, obras de arte, obras musicales, películas, organizaciones, empresas, especies, enfermedades, etc., tomadas de las versiones de Wikipedia en 35 lenguas⁵¹. Además de esto, DBpedia tiene versiones en 119 lenguas, si bien muchos conceptos se solapan con los de la versión en lengua inglesa, que resulta ser la más completa.

Es un hecho obvio que esta cantidad de entidades, ya clasificadas de manera jerárquica, resulta de extremo interés para el Onomasticón de FunGramKB. Asimismo, otra de las características interesantes de DBpedia es que su base de conocimiento enriquece la información extraída de Wikipedia –particularmente de las fichas informativas de cada artículo– con datos procedentes de otras fuentes, a través de 27,6

⁵¹ Inglés, alemán, francés, español, italiano, portugués, polaco, sueco, neerlandés, japonés, chino, ruso, finés, noruego, catalán, ucraniano, turco, checo, húngaro, rumano, volapük, esperanto, danés, eslovaco, indonesio, árabe, coreano, hebreo, lituano, vietnamita, esloveno, serbio, búlgaro, estonio y galés.

millones de enlaces a páginas externas y 45 millones de vínculos RDF a otras fuentes web de datos, lo que combinado con todas las fuentes de datos resulta en la astronómica cifra de 2,46 billones de ítems de información. Esto contribuye, naturalmente, a que la información almacenada en DBpedia sea, además de más completa que la disponible en Wikipedia, constantemente actualizada a la vez que las fuentes hacen lo propio, ya que DBpedia mantiene actualizada no sólo aquella información que extrae de Wikipedia, sino también la que proviene del resto de fuentes. Es más, los posteriores desarrollos de DBpedia con respecto a su inicio en 2007 permiten hoy una actualización de datos procedentes de Wikipedia en tiempo real, con un retraso mínimo de apenas unos pocos minutos, a través del proceso denominado *Live extraction*⁵² (Morsey *et al.*, 2012).

Quizás una de las aportaciones más valiosas que DBpedia puede ofrecerle al Onomasticón de FunGramKB sea la enumeración de propiedades de cada entidad para que después dichas propiedades puedan trasladarse a FunGramKB. Si se observa el proceso de población del módulo conceptual Ontología de FunGramKB, éste se realiza de una forma asistida a partir del conocimiento almacenado en diversos recursos lexicográficos (p. ej. diccionarios, tesoros de la lengua, corpora, etc.), así como también es asistida la posterior toma de decisiones que permite vincular en un mismo concepto diversas unidades léxicas. Ahora bien, y regresando al concepto de *proto*-estructuras (cf. sección 3.3), estas representaciones conceptuales de la Ontología responden a una realidad prototípica y atemporal. Esto significa que dichas representaciones conceptuales, en forma de postulados de significado, son en su gran mayoría propiedades estáticas que aluden a un prototipo no concreto de la realidad, sin perjuicio de que puedan contar en bastantes ocasiones con la etiqueta de “rebatibles”. Sin

⁵² Este proceso permite que la base DBpedia esté abierta para las actualizaciones constantemente a la vez que éstas suceden en Wikipedia, dando prioridad a aquellas páginas que sufren cambios de manera más frecuente sobre las que no han sido modificadas en mucho tiempo.

embargo, los valores de algunas propiedades de las entidades del Onomasticón, puesto que dichas entidades son instanciaciones concretas de la realidad o *bio*-estructuras, tienen la posibilidad de variar en el tiempo sin afectar a la identificación de la entidad como tal. Las entidades del Onomasticón, por tanto, parecen poseer una mayor capacidad de dinamismo, al tratarse de entidades reales y, en numerosos casos, vivas o existentes, lo que apunta a que la etiqueta de *bio* no es casual. A modo de ilustración, tomemos el concepto +EAT_00 de la Ontología de FunGramKB. En (6) se muestra el marco temático del concepto, así como su postulado de significado:

(6)

CONCEPTO: +EAT_00

MARCO TEMÁTICO:

(x1: +HUMAN_00 ^ +ANIMAL_00)Agent

(x2: +SOLID_00 ^ +LIQUID_00)Theme

(x3)Location

(x4)Origin

(x5: +STOMACH_00)Goal

POSTULADO DE SIGNIFICADO:

+(e1: +INGEST_00 (x1)Agent (x2)Theme (x3)Location (x4)Origin (x5)Goal

(f1: +MOUTH_00)Means (f2)Instrument (f3: (e2: +CHEW_00 (x1)Theme

(x2)Referent))Manner)

El ejemplo (6) muestra que hay ciertos participantes que intervienen en la descripción del concepto, lo que se indica en el marco temático (*Thematic Frame*). Por ejemplo, según reflejan los esquemas COREL en (6), es característico que el agente que ingiera el alimento, (x1)Agent, sea un humano o un animal y se requiere que lo haga llegar a su estómago (meta) (x5)Goal, a través de la boca (medio). La necesidad de que el participante que actúa como agente de la acción sea *animal* o *humano* es una característica prototípica en el mundo que conocemos, razón por la cual ambos

conceptos conforman el marco temático del concepto +EAT_00. Sin embargo, si tomamos una entidad del Onomasticón como, por ejemplo, una ciudad determinada, la cual posee propiedades como su número de habitantes, su extensión, su alcalde, etc., dichos valores podrían variar en el tiempo sin significar la desaparición de la entidad ni un cambio imposible en el mundo que conocemos. Por tanto, las bio-estructuras son capaces de tener una evolución, lo que las diferencia de las proto-microestructuras que componen la Ontología. Podría argumentarse, no obstante, que algunas de las propiedades de las estructuras del Onomasticón son también invariables, como por ejemplo aquellos eventos acaecidos en el pasado en las *historias*, lo que haría necesario concluir que, si bien las entidades del Onomasticón poseen algunas propiedades invariables, también poseen numerosas propiedades susceptibles de evolución, mientras que las estructuras de la Ontología suelen estar semánticamente más *fosilizadas*. No obstante, no podemos descartar que una posible evolución de los conceptos del mundo tal y como lo conocemos se reflejara en dichas estructuras, lo que podría llegar a manifestarse en la Ontología de FunGramKB tanto en los conceptos básicos como en los terminales, si bien en los primeros conllevaría un proceso mucho más costoso que en los segundos.

Retomando de nuevo la base DBpedia, otro de los principales valores que ésta ofrece a FunGramKB para la población del Onomasticón consiste en la estructuración de los datos que aquella posee. DBpedia almacena los datos procedentes de las fichas informativas de Wikipedia acerca de las entidades, completa esta información mediante el uso de otros recursos y, lo que resulta de mayor interés para FunGramKB, codifica de manera estructurada esta información. En el Capítulo IV del presente trabajo se describirá de manera detallada la metodología mediante la cual se crean reglas dentro de

FunGramKB que permiten importar esta información estructurada desde DBpedia para poblar el Onomasticón.

Resulta obvio el hecho de que, mediante esta importación a FunGramKB de datos procedentes de DBpedia, se está contribuyendo a la reusabilidad de recursos. Esto es un fenómeno de evidente importancia, ya que permite tanto la optimización de dichos recursos y la explotación de la eficiencia del modelo de origen, como la adaptación de la arquitectura de la plataforma de destino, puesto que desde FunGramKB se comparte la filosofía de que *“The usefulness of a knowledge base increases with more (correct) links to other knowledge bases (network effect), since this allows applications to combine information from several knowledge bases.”* (Auer y Lehmann, 2010:99). Igualmente, se podría considerar que Wikipedia, un recurso que posee tal ingente cantidad de información, debería ser reutilizado para ser codificado de manera más semántica, una tarea que se puede considerar que intentamos acometer a través de los valores contenidos en las estructuras de DBpedia. No en vano, el propio proyecto DBpedia reconoce el potencial y reutiliza la información de Wikipedia junto a la de otras muchas fuentes, incluidas YAGO y YAGO2. A modo ilustrativo, se puede observar la siguiente imagen que muestra todas las fuentes que DBpedia incorpora en sus estructuras, información que podrá ser a su vez integrada en FunGramKB cuando se haya completado la población semi-automática del Onomasticón con los datos procedentes de DBpedia:

cantidad de información que puede introducirse en la base de conocimiento mediante procesos semi-automáticos.

Asimismo, encontramos también proyectos similares al Onomasticón de FunGramKB en su concepción básica de aprovechar esta información estructurada y jerarquizada que DBpedia ofrece para utilizarla en posteriores desarrollos. El sistema PHYTHIA (Unger y Cimiano, 2011) presenta una similitud especial con FunGramKB, en tanto en cuanto proclama ser capaz de convertir cualquier *input* en lenguaje natural a una interlengua o esquema conceptual, un fenómeno que entraña cierto paralelismo con la noción de *Conceptual Logical Structure* –CLS– de FunGramKB que mencionábamos en la sección 3.1 de este capítulo. En PYTHIA, este esquema conceptual se transforma en una representación de significado y es posteriormente traducida a un lenguaje entendible por la máquina (por ejemplo, FLogic o SPARQL⁵⁴), lo que permite una consulta formal de la información que se desee obtener de la base de conocimiento. La Figura 7 resume la arquitectura global del sistema de consulta de PYTHIA:

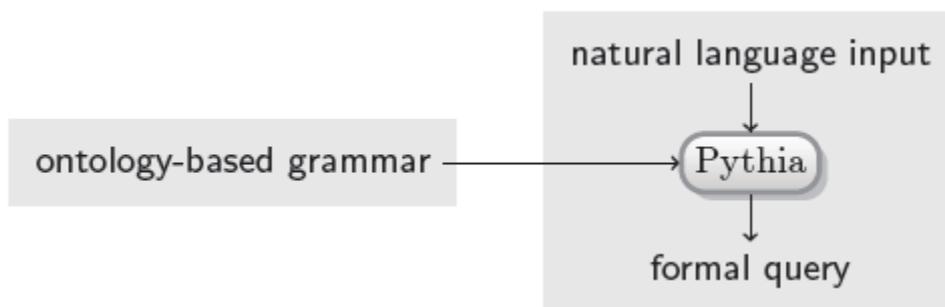


Figura 7. Arquitectura básica del sistema PHYTHIA (Unger y Cimiano, 2011:154).

Igualmente, llama la atención el hecho de que PYTHIA reivindique tener una profunda base lingüística, ya que para realizar la representación sintáctica el sistema se

⁵⁴ SPARQL es un lenguaje estandarizado de consulta de RDF. Para una información ampliada acerca de SPARQL, puede consultarse el siguiente enlace del W3C: <http://www.w3.org/TR/rdf-sparql-query>

basa en esquemas de la denominada LTAG (*Lexicalized Tree Adjoining Grammar*), mientras que para la representación semántica utilizan DUDEs, un tipo de estructuras de naturaleza denominada UDRS (*Underspecified Discourse Representation Structures*) aumentadas con información que permite una composición semántica más flexible (Unger y Cimiano, 2011). Asimismo, el sistema PYTHIA utiliza la herramienta LexInfo⁵⁵ (Cimiano *et al.*, 2011) para enriquecer la ontología con información acerca de la instanciación léxica, la cual ofrece un marco genérico para crear especificaciones declarativas de la interfaz léxico-ontológica mediante la conexión de los conceptos ontológicos y sus realizaciones lingüísticas (Unger y Cimiano, 2011). Este enlace que considera, por un lado, los conceptos y, por otro, sus instancias lingüísticas nos resulta muy familiar con respecto a la manera en que FunGramKB organiza el conocimiento en su Ontología, si bien la principal diferencia que encontramos frente a PYTHIA es que FunGramKB dedica diferentes módulos específicos a diferentes tipos de información conceptual y lingüística (Ontología, Cognición, Onomasticón, Lexicón, Gramaticón, etc.), mientras que PYTHIA parece aglutinar en un mismo conjunto tanto las lexicalizaciones de los conceptos como la información acerca de su morfología, los marcos de subcategorización y cómo los argumentos sintácticos y semánticos se corresponden entre ellos.

No obstante, estimamos que este espíritu unificador entre computación, lingüística e ingeniería del conocimiento constituye un fenómeno muy positivo, dado que en todo momento los autores de PYTHIA y LexInfo reivindican un papel más implicado de la lingüística en el desarrollo de herramientas ontológicas: *“Our main contribution is the model itself, but even more importantly a clear motivation why more*

⁵⁵ <http://lexinfo.net>

elaborate models for associating linguistic information with ontologies are needed.”

(Cimiano *et al.*, 2011:29).

El sistema PYTHIA, al igual que el Onomasticón de FunGramKB, utiliza la información contenida en las fichas informativas de Wikipedia a través del proyecto DBpedia, lo que consideramos una prueba más de la utilidad que dichos elementos aportan y de la voluntad global de reutilizar recursos ontológicos. Asimismo, PYTHIA menciona en todo momento su intención de contribuir a la construcción de la Web Semántica.

Sin embargo, la diferencia global que encontramos entre PYTHIA y FunGramKB parece recaer en el planteamiento de la utilidad del modelo: si bien PYTHIA se diseña como un sistema de consulta y respuesta, el Onomasticón de FunGramKB pretende encuadrarse dentro de un sistema global de PLN, lo que difiere de PYTHIA en cuanto que ésta parece estar algo más orientada a la minería de datos y su vínculo con la Web Semántica y algo menos al PLN, aunque la manera en que el sistema permite el *input* en lenguaje natural y lo transforma en una representación conceptual vinculada a conceptos ontológicos nos parece una loable tarea de PLN.

En el presente capítulo se ha presentado tanto la base de conocimiento FunGramKB y sus características como el lenguaje de representación que ésta utiliza (COREL), así como el formato que adquieren los postulados de significado en FunGramKB. De manera especial, se ha detallado el módulo conceptual Onomasticón, ya que es dentro de este módulo donde tiene lugar la discusión en el siguiente capítulo. Para efectuar con detalle la descripción del Onomasticón, se han visto las principales

fuentes y modelos utilizados para su población semi-automática (i.e. YAGO, YAGO2 y DBpedia), así como las semejanzas y diferencias que éstas presentan con FunGramKB.

A continuación, en el Capítulo IV se procederá a entrar de lleno en el primer objetivo de esta tesis doctoral, es decir, en la descripción y el análisis del proceso de creación de reglas que permiten la población semi-automática del Onomasticón y que, en última instancia, pueden permitir la proyección e integración en FunGramKB de la información y el conocimiento de las fuentes externas arriba mencionadas.

CAPÍTULO IV

METODOLOGÍA DE POBLACIÓN DEL ONOMASTICÓN

4.1. PLANTEAMIENTO PARA LA METODOLOGÍA

Como se ha mencionado en anteriores capítulos, la reusabilidad de las herramientas y modelos ontológicos y de PLN es una característica clave a la hora de optimizar recursos y maximizar los resultados obtenidos mediante su empleo, a la vez que dicha reusabilidad puede potenciar un uso automático de las mismas que permita ambos efectos.

Sin perder de vista este planteamiento, y con el fin de acometer el primero de los objetivos de la presente tesis doctoral, en el actual capítulo describimos la metodología mediante la cual el Onomasticón de FunGramKB es provisto de contenidos con ayuda de otras herramientas y modelos ontológicos de manera semi-automática, mediante un proceso que se divide en tres pasos (Periñán Pascual y Carrión Varela, 2011):

“i) Fase de construcción: a través de un editor en FunGramKB Suite, se construyen plantillas COREL que permitan plasmar el conocimiento almacenado en DBpedia.

ii) Fase de proyección: a partir de estas plantillas, se generan los retratos y las historias del Onomasticón.

iii) Fase de actualización: ya que las fuentes de conocimiento evolucionan, los esquemas conceptuales del Onomasticón también podrán ser actualizados.”

(Periñán Pascual y Carrión Varela, 2011:94)

Para comenzar a describir en detalle el punto (i) es necesario, en primer lugar, mostrar la estructuración de las propiedades que se confieren a las entidades en DBpedia. Con posterioridad a conocer la estructuración que presentan estas propiedades, describimos el proceso que comienza mediante la elección de cada una de ellas y la subsiguiente codificación en lenguaje COREL del concepto que evocan, así como los aspectos más significativos a los que se ha de prestar atención durante todo el proceso.

4.2. LAS PROPIEDADES EN DBPEDIA: LA JERARQUÍA DE CLASES

Como hemos visto en el Capítulo III, la base de conocimiento DBpedia almacena información clave acerca de entidades, tomada principalmente de Wikipedia y almacenada de manera jerárquica. Esta jerarquía de clases se organiza en la llamada *Ontología de DBpedia*. En ella, todas las clases surgen de una sola categoría llamada “Thing”, superordinado del cual parten todas las demás. La siguiente imagen muestra un fragmento de la vasta jerarquía de clases de la ontología de DBpedia:

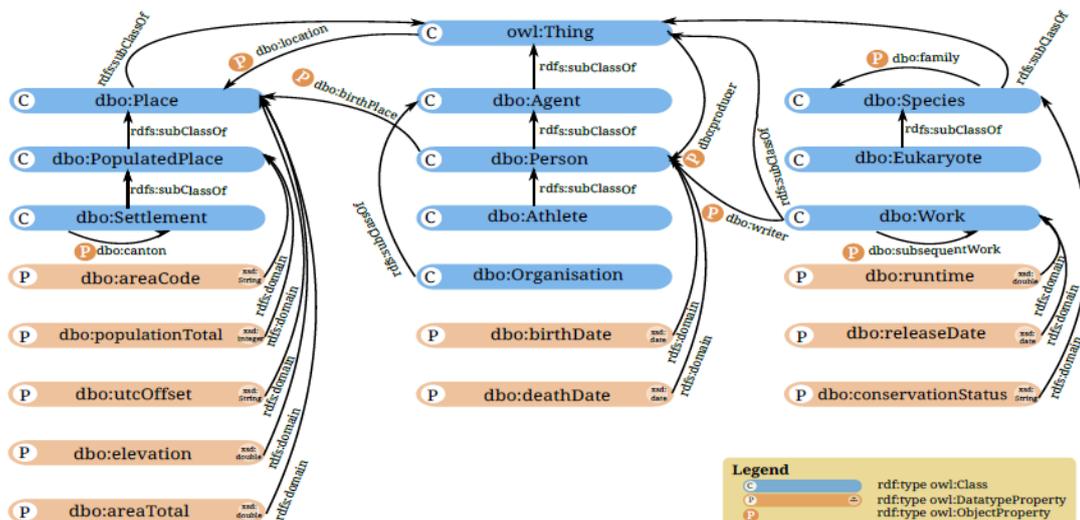


Figura 8. Fragmento de la Ontología de DBpedia (Lehmann *et al.*, en prensa).

La jerarquía de clases de la Ontología de DBpedia es una extensísima red de más de 320 clases de entidades definidas por un total de 1.650 propiedades vinculadas a dichas clases y que definen cada instancia incluida en ellas (Lehmann *et al.*, en prensa). Por ejemplo, si tomamos la clase “Place” de DBpedia, encontramos un gran número de entidades dentro de ella, que a la vez están organizadas en lo que DBpedia denomina “*subClassOf*”, es decir, sub-clases más concretas que aglutinan los diferentes tipos de lugar que se pueden identificar dentro de la clase “Place”, la cual actúa de hiperónimo (p. ej., “Populated Place”, “Building”, “Body of Water”, etc.). Esta jerarquía podemos contemplarla en la siguiente ilustración:

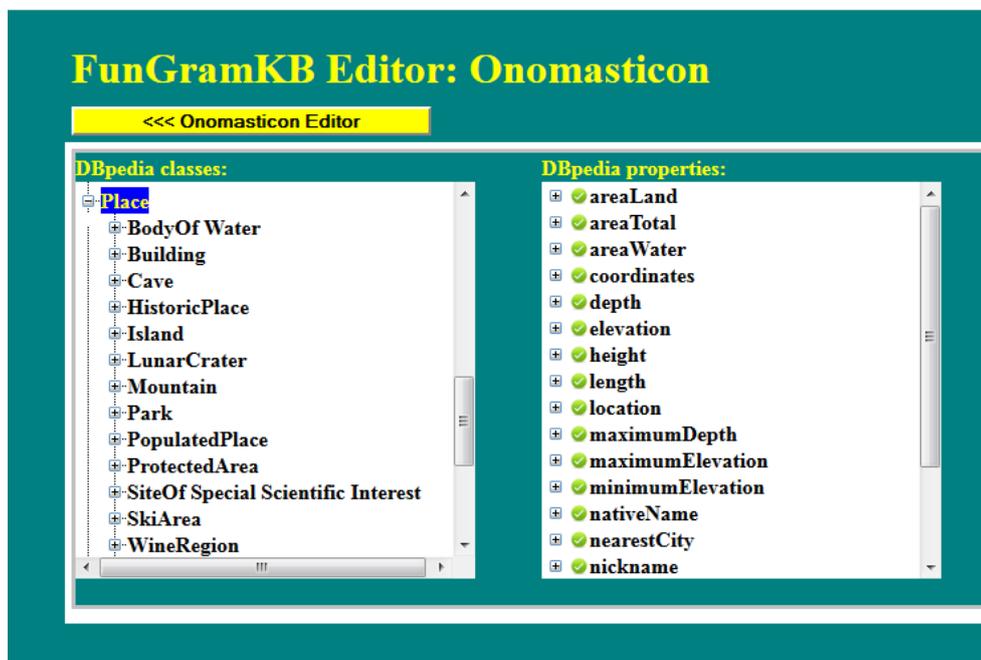


Ilustración 3. La clase “Place” y sus diversas sub-clases y propiedades en el Onomasticon de FunGramKB.

Con carácter general, las entidades que se encuentran aglutinadas dentro de la misma clase o sub-clase tienden a compartir una serie de rasgos o características que las definen, las cuales denominamos *propiedades*, y cuyo valor es obtenido de la información proyectada desde las fichas informativas de Wikipedia a través de DBpedia. Por ejemplo, una propiedad de las entidades que componen la clase “Place”

puede ser la altitud a la que un lugar se encuentra, la profundidad que posee o el área total que ocupa. Estas propiedades obtienen un valor según los datos que aporta la ficha informativa correspondiente de Wikipedia, por lo que cada entidad tendrá un valor asignado a cada una de sus propiedades (p. ej., los metros de altitud o de profundidad a la que se encuentra o el número de metros cuadrados que ocupa su área total). Si se accede a cada una de estas propiedades a través de DBpedia, se obtiene más información acerca de ellas, como puede ser una breve definición de las mismas, alguna reseña aclaratoria (“*comments*”) o la unidad de medida, si procede, en la que se expresa su valor. Esta unidad de medida resultará relevante a la hora de crear de manera asistida la regla que definirá dicha propiedad dentro del Onomasticón de FunGramKB, algo que más adelante ponemos de manifiesto en la descripción del proceso de creación de reglas. A continuación mostramos, a modo de ejemplo, las imágenes que ilustran los datos que DBpedia proporciona acerca de tres propiedades pertenecientes a la clase “Place”, i.e. *elevation*, *areaTotal* y *nearestCity*:

About: [elevation \(m\)](#)
An Entity of Type : [DatatypeProperty](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

average elevation above the sea level

Property	Value
rdfs:type	owl:DatatypeProperty
rdfs:comment	average elevation above the sea level
rdfs:domain	dbpedia-owl:Place
rdfs:label	elevation (m)
rdfs:range	xsd:double

Browse using: [OpenLink Data Explorer](#) | [Zitgist Data Viewer](#) | [Marbles](#) | [DISCO](#) | [Tabulator](#) Raw Data in: [N3/Turtle](#) | [JSON+RDF](#) | [RDF/XML](#) | [About](#)

POWERED BY VIRTUOSO LINKINGOPENDATA W3C SPARQL OPEN DATA

Ilustración 4. Propiedad *elevation*, dentro de la clase “Place”⁵⁶

⁵⁶ <http://dbpedia.org/ontology/elevation>, fecha de consulta 02/06/2010.

The screenshot shows the DBpedia page for the property 'area total (m2)'. At the top, it says 'About: area total (m2)' and 'An Entity of Type: DatatypeProperty, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org'. Below this is a table with two columns: 'Property' and 'Value'. The table lists several properties and their corresponding values:

Property	Value
rdf:type	owl:DatatypeProperty
rdfs:domain	dbpedia-owl:Place
rdfs:label	area total (m2)
rdfs:range	xsd:double
owl:sameAs	http://sw.opencyc.org/concept/Mx8Ngh4rY_nOYBt-QdiS5seAT9e7DQ-lahR0cDovL2RicGVkaWEub3JnL29udG9sb2d5L2FyZWVub3RhbA

At the bottom, there are links for 'Browse using' (OpenLink Data Explorer, Zitgist Data Viewer, Marbles, DISCO, Tabulator) and 'Raw Data in' (N3/Turtle, JSON+RDF, RDF/XML, OData/Atom, About). There are also logos for 'POWERED BY VIRTUOSO', 'LINKING OPENDATA', 'W3C SPARQL', and 'OPEN DATA'.

Ilustración 5. Propiedad *areaTotal*, dentro de la clase “Place”⁵⁷

The screenshot shows the DBpedia page for the property 'nearest city'. At the top, it says 'About: nearest city' and 'An Entity of Type: ObjectProperty, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org'. Below this is a table with two columns: 'Property' and 'Value'. The table lists several properties and their corresponding values:

Property	Value
rdf:type	owl:ObjectProperty
rdfs:domain	dbpedia-owl:Place
rdfs:label	nearest city
rdfs:range	dbpedia-owl:PopulatedPlace

At the bottom, there are links for 'Browse using' (OpenLink Data Explorer, Zitgist Data Viewer, Marbles, DISCO, Tabulator) and 'Raw Data in' (N3/Turtle, JSON+RDF, RDF/XML, OData/Atom, About). There are also logos for 'POWERED BY VIRTUOSO', 'LINKING OPENDATA', 'W3C SPARQL', and 'OPEN DATA'.

Ilustración 6. Propiedad *nearestCity*, dentro de la clase “Place”⁵⁸

Como se observa en la Ilustración 4, Ilustración 5 e Ilustración 6, cada propiedad puede expresar su valor en una unidad de medida diferente, además de que, por otra parte, DBpedia puede indicar desigual cantidad de información acerca de cada una de ellas. En el caso de *elevation*, se aprecia en la Ilustración 4 una breve descripción o aclaración de la propiedad en forma de comentario (“*average elevation above the sea level*”), además de indicarse que esta propiedad se encuentra dentro de la clase “Place” (“*domain*”) y que su valor o unidad de medida (denominado “*label*” en DBpedia) vendrá expresado en metros (m). Por otro lado, en la Ilustración 5, correspondiente a la propiedad *areaTotal*, observamos que, si bien se mantiene la indicación de la clase “Place”, desaparecen los comentarios aclaratorios, aunque la unidad de medida se sigue mostrando. En este caso, se indica que los valores mostrados por esta propiedad (“*label*”) se expresarán en metros cuadrados (m²). Finalmente, si se presta atención a la propiedad *nearestCity*, mostrada en la Ilustración 6, no aparece ninguna unidad de

⁵⁷ <http://dbpedia.org/ontology/areaTotal>, fecha de consulta 02/06/2010.

⁵⁸ <http://dbpedia.org/ontology/nearestCity>, fecha de consulta 02/06/2010.

medida dentro de un sistema o clasificación internacional, puesto que esta propiedad carece de ella, a pesar de que se indica que el valor vendrá expresado con un topónimo, el cual será una entidad nombrada de las que se encuentran en Wikipedia. Cuando la indicación del tipo de valor que se asigna a una propiedad no constituye una unidad de medida, sino otro tipo de concepto, se puede llegar a identificar en DBpedia mediante el término “*range*”, como complemento a la etiqueta “*label*” que hemos visto arriba y que se suele aplicar a unidades de medida numérica. No obstante, a pesar de que en una propiedad no encontremos la etiqueta “*label*” o unidad de medida definida, en ciertas ocasiones el propio contenido de la propiedad puede hacer necesario que debamos indicar con más precisión el tipo de entidad o “*range*” que representará a esa propiedad (por ejemplo, si se trata de una fecha, un lugar, etc.). Toda esta información añadida acerca de las propiedades de una clase es de gran importancia para su posterior codificación en lenguaje COREL, ya que aporta datos aclaratorios que nos sirven de ayuda para la creación de las reglas de cada propiedad.

4.3. LAS FASES DEL PROCESO DE CREACIÓN DE REGLAS

Al comienzo de este capítulo hemos enumerado brevemente las fases que componen el protocolo de creación de las reglas que nos permiten efectuar el trasvase automático de información procedente de DBpedia para insertarla en el Onomasticón de FunGramKB. Este proceso, compuesto de tres fases –fase de construcción, fase de proyección y fase de actualización (Periñán Pascual y Carrión Varela, 2011) – es detallado a continuación de manera exhaustiva y mediante la aportación de diversos ejemplos que ilustran cada una de ellas. Esta metodología, si bien fue inicialmente esbozada en Carrión Varela (2010) y redefinida en Periñán Pascual y Carrión Varela (2011), en el presente trabajo se desarrolla un paso más, ya que aportamos un análisis más profundo tanto del proceso de creación de reglas como de los resultados obtenidos.

En el presente capítulo nos ocupamos de la descripción y el análisis de la metodología de creación de reglas para ceder enteramente el Capítulo V al análisis de los resultados obtenidos.

4.3.1. LA FASE DE CONSTRUCCIÓN

La *fase de construcción* tiene como finalidad la elaboración de manera asistida de reglas basadas en plantillas que permitan extraer, mediante esquemas COREL, el conocimiento almacenado en DBpedia y proyectarlo al Onomasticón de FunGramKB.

Los pasos, a grandes rasgos, que se pueden enumerar dentro del protocolo para la creación asistida de las reglas son los siguientes (cf. Perrián Pascual y Carrión Varela, 2011):

- 1) Relevancia y pertinencia de la propiedad para FunGramKB.
- 2) Identificación del tipo de bio-estructura (i.e. *retrato* o *historia*) a la que pertenecerá la predicación.
- 3) En caso de valor numérico, identificación de la unidad de medida en la que se expresa el valor de la propiedad.
- 4) Construcción de la plantilla mediante la elaboración de las predicaciones en lenguaje COREL.

Una vez realizado el esquema de la metodología mediante la enumeración de los pasos que componen el procedimiento, procedemos a describir en detalle los protocolos seguidos en cada uno de ellos para completar la creación de reglas en forma de plantillas COREL.

4.3.1.1. Relevancia y pertinencia de la propiedad para FunGramKB

Antes de crear una regla para cualquier propiedad, es necesario realizar un ejercicio previo que determine, dentro del listado de propiedades que DBpedia

proporciona para cada clase o subclase⁵⁹, cuáles de ellas resultan pertinentes y relevantes para FunGramKB y cuáles no. La relevancia conferida a cada propiedad puede relacionarse con diversos factores: (i) su reusabilidad y pertinencia para el PLN, (ii) su grado de generalidad o prototipicidad, (iii) su capacidad expresiva o (iv) motivos técnicos de la herramienta utilizada. Por ejemplo, a pesar de que FunGramKB fomenta la incorporación de diferentes conceptos culturales, a través de la inclusión de entidades nombradas propias de una cultura, en algunos casos existen propiedades que no desempeñan ningún papel en los procesos lingüísticos que tienen lugar en un sistema de PLN. Un ejemplo de esta ausencia de productividad para el PLN lo encontramos en las siguientes propiedades pertenecientes a la clase “City”⁶⁰:

distanceToBelfast
distanceToCardiff
distanceToDouglas
distanceToEdinburgh
distanceToLondon
gaelicName
federalState
frazioni
gaelicName
irishName
manxName
scottishName
welshName

Tabla 2. Lista de propiedades de “City” con ausencia de productividad semántica y conceptual.

En el ejemplo que muestra la Tabla 2, observamos que los motivos por los cuales se ha decidido prescindir de la creación de una regla para las propiedades listadas responden a diversas causas. Así pues, estimamos que no es necesaria la creación de una regla para reflejar las propiedades que denotan el nombre de una ciudad en una lengua regional británica (p. ej., *gaelicName*, *irishName*, *manxName*, *scottishName*,

⁵⁹ En la Ilustración 3 del presente capítulo se ha mostrado la imagen que presenta en el Onomasticón una clase y su correspondiente lista de propiedades.

⁶⁰ Ruta conceptual de “City”: Place > populatedPlace > City.

welshName), ya que dichas traducciones se pueden considerar un nombre alternativo de una misma entidad. La herramienta de edición FunGramKB Suite del Onomasticón posee una opción denominada “*alternative name*”, que permite indicar las diferentes designaciones de una entidad como nombres alternativos de la misma, por lo que el sistema sería capaz de redireccionar a la misma entidad, sin necesidad de elaborar una regla para ello. Además, FunGramKB no trabaja con estas lenguas regionales, por lo que, igualmente, no sería posible incorporar esta información en un postulado de significado.

Tampoco se ha creado una regla para aquellas propiedades que indican la distancia entre cualquier ciudad y las ciudades de Belfast, Cardiff, Douglas, Edimburgo o Londres, ya que el objetivo de estas propiedades responde a otros motivos computacionales enfocados a la resolución de problemas prácticos (p. ej. ubicación espacial de un usuario) y no a motivos basados en necesidades de los procesos lingüísticos de PLN.

Otro ejemplo de propiedades cuya naturaleza no resulta relevante para los procesos de PLN son las propiedades *frazioni* y *federalState*, las cuales responden a características de las demarcaciones geográfico-políticas de los municipios de Italia y de Estados Unidos, respectivamente, por lo que no consideramos de utilidad para nuestros objetivos su inclusión en el Onomasticón.

Finalmente, debemos destacar que existe otro tipo de propiedades cuya función definitoria de las entidades a las que se atribuyen parece más orientada a tareas de minería de datos que a PLN, ya que se limitan a reflejar atributos estadísticos o puramente numéricos de mermada trascendencia para el PLN, aunque sí la tendrían para dichas tareas de minería de datos. Un ejemplo de este tipo de propiedades lo

encontramos en *populationDensity* o *populationMetroDensity*, pertenecientes a la clase “Populated Place” o la propiedad *coordinates*, perteneciente a “Place”. Este tipo de propiedades, cuya productividad ontológica es bastante reducida, no constituyen un objetivo al amparo del presente trabajo, por lo que no se ha procedido a elaborar una regla para ellas. Asimismo, también consideramos que propiedades como *populationDensity* o *populationMetroDensity* no constituyen una referencia habitual por parte de un hablante común nativo, debido a su carácter más especializado o técnico, lo que seguramente las haría merecedoras de ser incluidas en una ontología terminológica especializada (por ejemplo, de terminología estadística o demográfica). En el Apéndice I del presente trabajo se incluye el inventario completo de las clases y subclases en las que se ha trabajado, así como la totalidad de las reglas creadas y una indicación de aquellas propiedades para las que se prescinde de creación de regla, identificadas como “n/r” (*no rule*).

4.3.1.2. Identificación del tipo de bio-estructura (i.e. retrato o historia) a la que pertenecerá la predicación

Tras la clasificación y descarte previo descrito en el paso anterior, se ha de proseguir con el siguiente eslabón de la cadena que permite crear la regla. A modo ilustrativo, retomaremos la propiedad *elevation* mostrada anteriormente, que hace referencia a una característica geográfica de las entidades que engloba la clase “Place”:

About: [elevation \(m\)](#)
An Entity of Type: [DatatypeProperty](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

average elevation above the sea level

Property	Value
rdf:type	owl:DatatypeProperty
rdfs:comment	average elevation above the sea level
rdfs:domain	dbpedia-owl:Place
rdfs:label	elevation (m)
rdfs:range	xsd:double

Browse using: [OpenLink Data Explorer](#) | [Zitgist Data Viewer](#) | [Marbles](#) | [DISCO](#) | [Tabulator](#) Raw Data in: [N3/Turtle](#) | [JSON+RDF](#) | [RDF/XML](#) | [About](#)

POWERED BY VIRTUOSO LINKINGOPENDATA W3C SPARQL OPEN DATA

Ilustración 7. Propiedad *elevation* dentro de la clase “Place”⁶¹

Como se observa en la Ilustración 7, gracias a la información que DBpedia aporta sobre la propiedad *elevation* podemos detectar que el valor de ésta se expresa en metros, por lo que ésa será la medida que se codificará en COREL a la hora de elaborar la regla, como veremos en el paso 3 del procedimiento. En segundo lugar, vemos que DBpedia aporta una breve reseña aclaratoria en el campo “comment” que, en este caso, se podría considerar una indicación relativa a la desambiguación de la propiedad *elevation*. Como resultado de esta desambiguación, se observa que la propiedad *elevation* en DBpedia se refiere al nivel promedio de altitud sobre el nivel del mar de un determinado lugar. Mediante esta aclaración se descarta que la propiedad llamada *elevation* en DBpedia se refiera a la altura, la cual es la distancia vertical entre dos puntos de la superficie terrestre y no a partir del nivel del mar. Asimismo, también se explicita mediante este breve comentario que el valor asignado a la propiedad *elevation* de una entidad será un valor promedio (“average”), puesto que debido a las diferencias geológicas y geográficas del relieve de un determinado lugar, es posible que en superficies de extensión amplia tengan lugar variaciones en altitud entre dos puntos de la zona. En caso de que la dimensión que se deseara reflejar fuese la altitud máxima o mínima de un lugar, DBpedia también cuenta con otras propiedades que reflejan estas

⁶¹ <http://dbpedia.org/ontology/elevation>, fecha de consulta 02/06/2010.

características de manera separada, i.e. las propiedades *maximumElevation* y *minimumElevation*.

Por otro lado, resulta interesante destacar que, para algunas propiedades, descubrimos que DBpedia nos facilita un enlace bajo la denominación “*equivalentProperty*” a la herramienta Schema.org⁶². Schema.org es un proyecto colaborativo en el cual se encuentra la lista de esquemas (por ejemplo, etiquetas HTML) más comúnmente utilizados por programadores de todo el mundo, esquemas que son reconocidos por la mayoría de los motores de búsqueda actuales (p. ej. Bing, Google, Yahoo! o Yandex). Estos motores utilizan dichas etiquetas para mejorar los resultados de las búsquedas realizadas por los usuarios, ya que aquéllas proporcionan información adicional a los textos que acompañan. Asimismo, a la hora de elaborar las reglas para el Onomasticón de FunGramKB, también nos resulta de gran utilidad aclaratoria el encontrar propiedades de DBpedia vinculadas a otras propiedades que aparecen listadas en Schema.org., ya sean etiquetas HTML u otro tipo de esquemas, puesto que, de este modo, podemos encontrar una explicación más extensa y detallada acerca de la referencia de una propiedad. A modo de ilustración, presentamos la propiedad *child* de DBpedia, asignada a la clase “Person”:

About: [child](#)
An Entity of Type: [ObjectProperty](#), from Named Graph: <http://dbpedia.org/resource/classes#>, within Data Space: [dbpedia.org](#)

Property	Value
rdf:type	<ul style="list-style-type: none">owl:ObjectProperty
rdfs:domain	<ul style="list-style-type: none">dbpedia-owl:Person
rdfs:label	<ul style="list-style-type: none">Kindchildkindkindkindkindkindkind
rdfs:range	<ul style="list-style-type: none">dbpedia-owl:Person
owl:equivalentProperty	<ul style="list-style-type: none">http://schema.org/children

Browse using: [OpenLink Data Explorer](#) | [OpenLink Faceted Browser](#) | Raw Data in: [CSV](#) | [RDF](#) ([N-Triples](#) | [N3/Turtle](#) | [JSON](#) | [XML](#)) | [OData](#) ([Atom](#) | [JSON](#)) | [Microdata](#) ([JSON](#) | [HTML](#)) | [JSON-LD](#) | [About](#)

POWERED BY VIRTUOSO | LINKING OPEN DATA | W3C SPARQL | OPEN DATA | W3C HTML + RDF

This content was extracted from [Wikipedia](#) and is licensed under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#)

Ilustración 8. Propiedad *child* dentro de la clase “Person”⁶³

⁶² <http://schema.org>

⁶³ <http://dbpedia.org/ontology/child>, fecha de consulta 04/03/2014.

En el presente ejemplo, a la luz de los datos que nos ofrece DBpedia, y a pesar de que el ingeniero del conocimiento que esté elaborando esta propiedad pueda suponer que *child* se refiere a los hijos que una persona tiene, no termina de manifestarse de manera clara a qué alude esta propiedad. Podría, por ejemplo, referirse a de quién es hijo la persona y no al revés, o simplemente tener un valor booleano frente a otra hipotética propiedad que se denominase *adult*. Sin embargo, en la

Ilustración 8 podemos observar que DBpedia ha efectuado el esfuerzo de vincular su propiedad *child* con otra propiedad equivalente en Schema.org denominada *children* y que viene explicada de este modo:

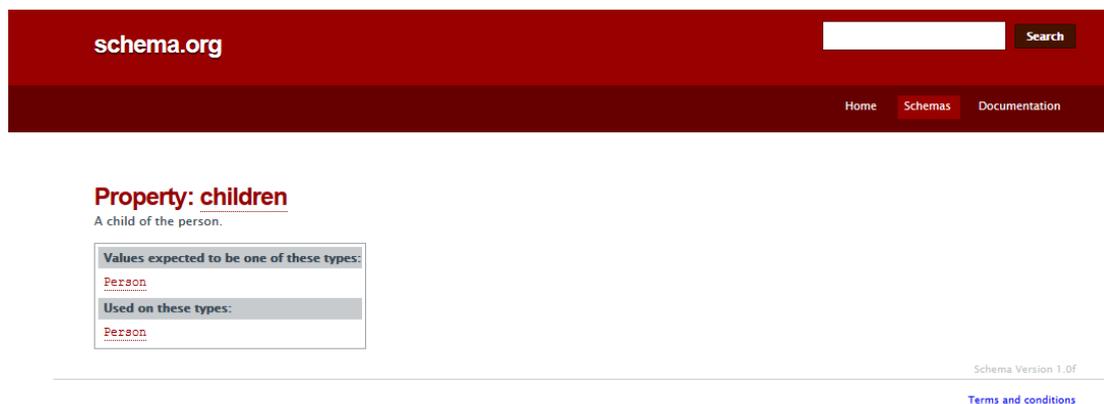


Ilustración 9. Propiedad *children* en Schema.org⁶⁴

Por tanto, gracias a esta información podemos confirmar que tanto la propiedad *child* de DBpedia como la propiedad *children* de Schema.org son equivalentes y que ambas aluden a los hijos que una persona tiene. De este modo, el ingeniero del conocimiento puede obtener una confirmación adicional de los datos aportados por DBpedia para una propiedad concreta y así elaborar de manera más segura la regla pertinente. Así pues, una vez que se ha realizado la observación pormenorizada de la información que DBpedia aporta acerca de la propiedad, con la ayuda del enlace a

⁶⁴ <http://schema.org/children>, fecha de consulta 04/03/2014.

Schema.org, podemos proceder a la identificación de la bio-estructura a la que pertenece la misma.

Como se ha mencionado en el capítulo anterior, las entidades o bio-estructuras del Onomasticón de FunGramKB pueden ser clasificadas como retratos (*snapshots*) o historias (*stories*), en función de si se trata de propiedades estáticas en un momento particular (retratos) o de propiedades que han de encajarse dentro de un esquema temporal determinado (historias). Por ende, existirán propiedades de DBpedia que pueden contribuir a la construcción o bien de retratos o bien de historias. En el ejemplo *elevation* mencionado con anterioridad, dicha propiedad formaría parte de un retrato, ya que el valor que esta propiedad adopta en relación a una entidad determinada es aplicable en diversas situaciones del continuum espacio-temporal y, por tanto, no es específica de un único punto temporal. Además, este valor puede estar sujeto a cambio (p. ej., si se considera el hipotético caso de que por causas naturales, como un terremoto, un corrimiento de tierra o la subida del nivel del mar, la altitud de un lugar sufra modificaciones).

Un ejemplo de propiedades que adoptan el perfil opuesto –historias– serían aquellas tales como *birthDate* o *deathDate*, que pueden encontrarse dentro de la clase “Person”⁶⁵ de DBpedia, ya que se trata de propiedades que han de ser encajadas de manera invariable dentro de una secuenciación temporal. Para indicar el tipo de bio-estructura a la que pertenece la propiedad, se debe seleccionar la opción correspondiente en la interfaz de edición del Onomasticón, que ilustramos en la figura que se incluye a continuación:

⁶⁵ <http://dbpedia.org/ontology/Person>

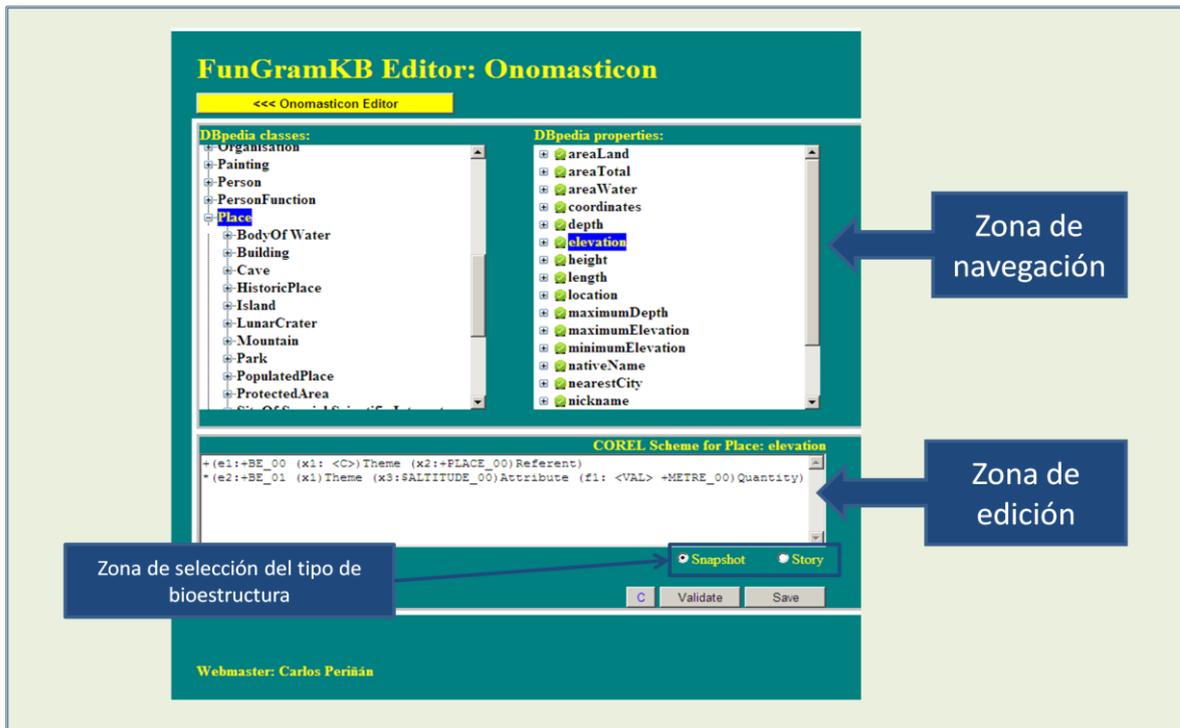


Figura 9. Interfaz del DBpedia Mapper en el editor del Onomasticon.

4.3.1.3. En caso de valor numérico, identificación de la unidad de medida en la que se expresa el valor de la propiedad

A continuación, al igual que se ha hecho en el paso 2, antes de proceder a elaborar la regla en lenguaje COREL es importante identificar los parámetros o medidas que se necesitará tener en cuenta a la hora de realizar la codificación. En el ejemplo que nos ocupa, gracias a la información que DBpedia aporta sobre la propiedad *elevation* (como se ha mostrado en la Ilustración 7), se puede identificar el metro como unidad de medida de los valores que *elevation* adquiere para cada entidad. Puesto que en la regla que elaboremos en lenguaje COREL se debe indicar esta unidad, en este momento procede efectuar una búsqueda en la Ontología de FunGramKB del concepto que se corresponde con la unidad que se necesita, ya que los conceptos usados en la regla en lenguaje COREL han de estar previamente incluidos en la Ontología de FunGramKB. En el actual ejemplo, buscaríamos en la Ontología de FunGramKB el concepto vinculado a la unidad léxica “metro”. Se trata, nuevamente, de otra manifestación de la

interrelación necesaria entre los diferentes módulos de FunGramKB, lo que contribuye a su consistencia, cohesión e interoperabilidad: no es posible utilizar ningún concepto en las reglas del Onomasticón que no exista previamente en la Ontología de FunGramKB. Con objeto de descubrir si la unidad léxica “metro” está ya incluida como concepto dentro de la Ontología de FunGramKB, se ha de recurrir a la herramienta de búsqueda dentro de dicho módulo. A tal efecto, y por razones prácticas de optimización de recursos, no es necesario acceder a la Ontología de FunGramKB de manera independiente, ya que se puede encontrar un acceso directo a la herramienta de búsqueda dentro de la aplicación en la que estamos trabajando, FunGramKB Editor del Onomasticón, como se muestra en la siguiente ilustración:

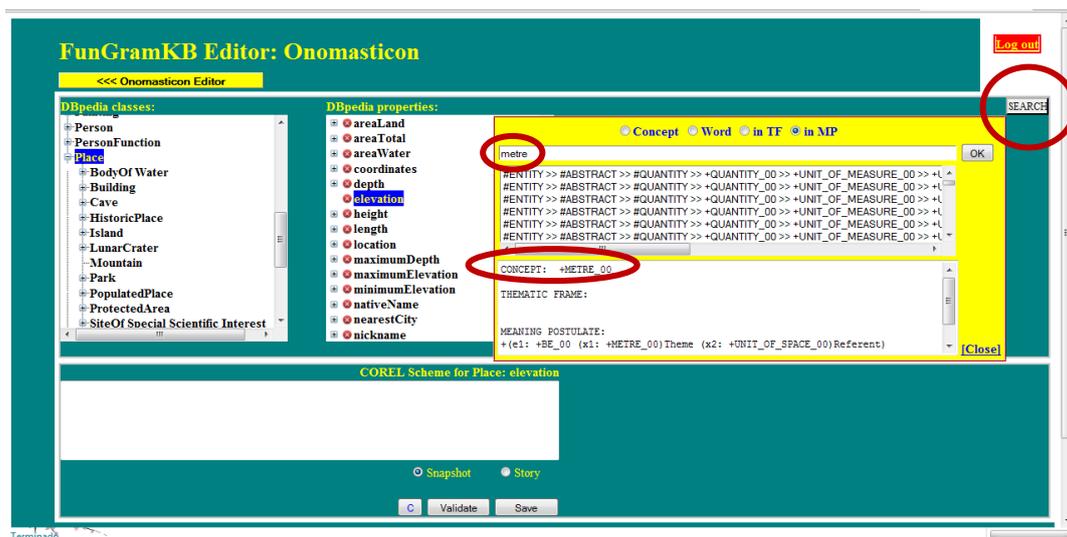


Ilustración 10. Búsqueda de la unidad léxica “metro” en FunGramKB Editor del Onomasticón.

En caso de haber realizado la búsqueda del concepto utilizando la unidad léxica “metro” en español, se habría llegado al mismo resultado, ya FunGramKB permite asignar diferentes lexicalizaciones en varias lenguas naturales a un mismo concepto expresado en lenguaje COREL. La Ilustración 11 muestra este fenómeno:

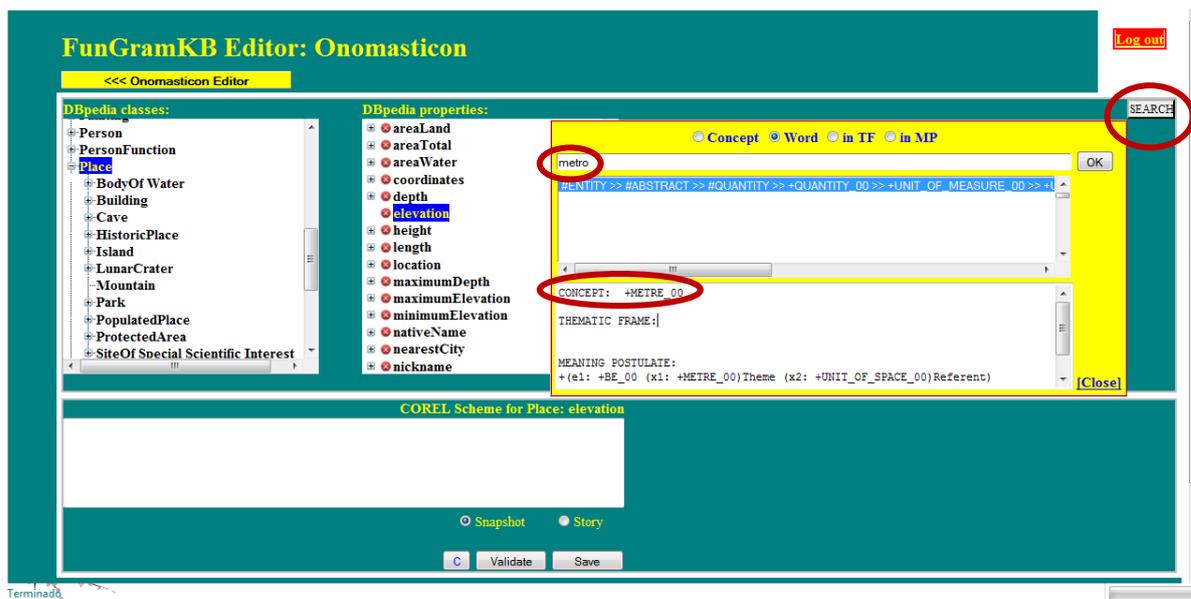


Ilustración 11. Búsqueda de la unidad léxica “metro” en *FunGramKB Editor* del Onomasticón.

En el hipotético –pero posible– caso de que la unidad de medida que se necesita emplear en la creación de la regla no apareciera en la búsqueda realizada, sería necesario crearla en la Ontología de FunGramKB antes de utilizarla en el Onomasticón. Esta tarea tiene como objetivo mantener la correlación entre módulos, ya que en las reglas introducidas en el Onomasticón no es posible emplear conceptos inexistentes en la Ontología de FunGramKB. Para llevar a cabo la tarea de introducción del nuevo concepto, se requiere acceder a la aplicación FunGramKB Editor de la Ontología de FunGramKB y crear dentro de este módulo el concepto terminal⁶⁶ que indica la unidad de medida, siguiendo el procedimiento pertinente a tal fin. No obstante, es importante recalcar que este paso no debe ser llevado a cabo sin antes realizar todas las comprobaciones posibles de que el concepto que se busca, efectivamente, no existe en la Ontología de FunGramKB.

Un ejemplo concreto de la inexistencia de un concepto necesario ha sido la unidad “m²”, que constituye la unidad de medida de la propiedad *areaTotal* mostrada en

⁶⁶ Puesto que el inventario de conceptos básicos es cerrado, el nuevo concepto a crear debe ser un concepto terminal, identificado con el símbolo \$.

la Ilustración 5. A la hora de elaborar la regla para esta propiedad, y tras efectuar las comprobaciones pertinentes en la Ontología de FunGramKB, no se ha encontrado dicha unidad de medida en su inventario de conceptos. Para crear el concepto “metro cuadrado” en la Ontología, al que se asignó la denominación en lenguaje COREL \$SQUARE_METRE_00, se han seguido los pasos que enumeramos a continuación:

- i) Consulta del concepto en diversos recursos lexicográficos: diccionarios, tesauros, etc. Los recursos lexicográficos consultados han sido el Diccionario de la Lengua Española de la RAE en su versión online⁶⁷, Longman Dictionary of Contemporary English Online⁶⁸, Collins English Dictionary en su versión online⁶⁹, Cambridge Dictionaries Online⁷⁰ y Merriam-Webster en su versión online⁷¹. Esta búsqueda ha arrojado numerosos resultados acerca de la definición de metro cuadrado, siendo los rasgos semánticos más comunes los que se pueden resumir como “*an area of measurement equal to a square with sides a metre long*”.
- ii) Una vez consultados los recursos lexicográficos y tomando como punto de partida las diversas definiciones encontradas, se procede a consultar la jerarquía de la Ontología de FunGramKB, con objeto de identificar el lugar en dicha jerarquía donde se debe introducir el nuevo concepto. En este caso, el concepto se identifica como una unidad de medida, lo cual se corresponde con el concepto básico +UNIT_OF_MEASURE_00 y, más concretamente, con uno de sus conceptos subordinados:

⁶⁷ <http://rae.es/obras-academicas/diccionarios/diccionario-de-la-lengua-espanola>

⁶⁸ <http://www.ldoceonline.com>

⁶⁹ <http://www.collinsdictionary.com/dictionary/english>

⁷⁰ <http://dictionary.cambridge.org>

⁷¹ <http://www.merriam-webster.com>

+UNIT_OF_SPACE_00, cuya ubicación en la Ontología ilustramos a continuación:

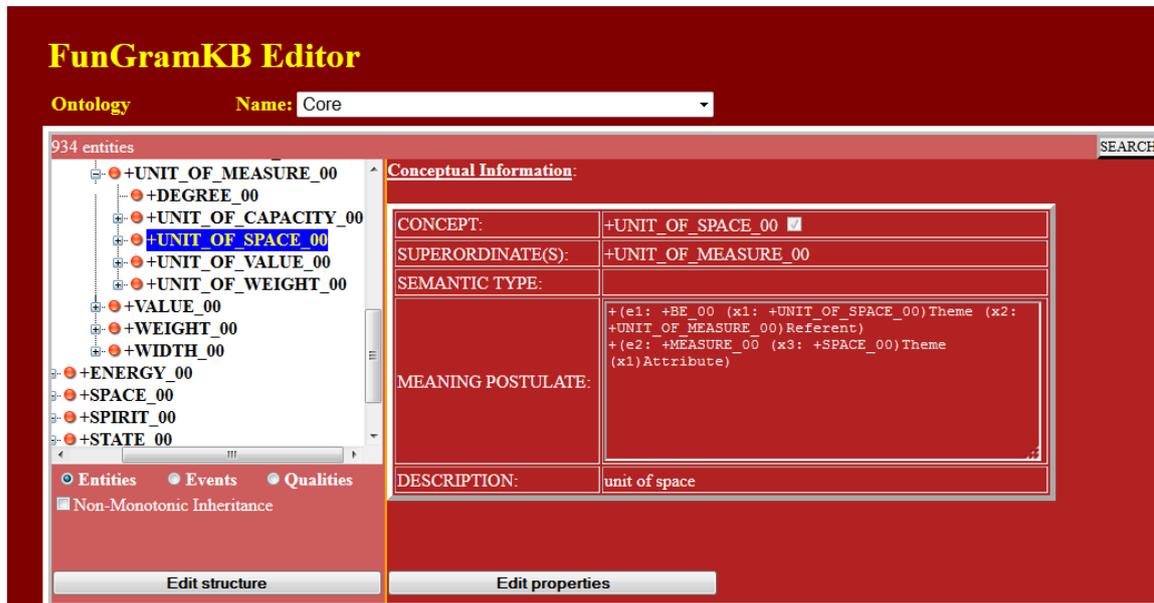


Ilustración 12. Jerarquía del concepto básico +UNIT_OF_SPACE_00.

iii) Una vez ubicado el lugar en la jerarquía, se procede a identificar los conceptos semánticos necesarios para comenzar a crear el postulado de significado del nuevo concepto. Puesto que todos los recursos lexicográficos mencionan de alguna forma las ideas de *medida*, *anchura*, *longitud*, *superficie* y la unidad de medida *metro*, se procede a efectuar una búsqueda en la Ontología de FunGramKB para localizar los posibles conceptos que reflejen el contenido semántico de estas unidades léxicas, siendo todos ellos localizados: +MEASURE_00, +WIDTH_00, +LENGTH_00, +SURFACE_00 y +METRE_00.

iv) Finalmente, procedemos a elaborar el postulado de significado del nuevo concepto \$SQUARE_METRE_00, cuya descripción en lenguaje natural es “*an area of measurement equal to a square with sides a metre long*” y

cuyo postulado de significado en lenguaje COREL indicamos a continuación:

Concepto	\$SQUARE_METRE_00
Postulado de significado	<p>+(e1: +BE_00 (x1: \$SQUARE_METRE_00)Theme (x2: +UNIT_OF_SPACE_00)Referent)</p> <p>*((e2: +MEASURE_00 (x3: +SURFACE_00)Theme (x1)Attribute)(e3: +BE_01 (x3)Theme (x4:+WIDTH_00)Attribute)(e4: +MEASURE_00 (x4)Theme (x5: 1 +METRE_00)Attribute))</p> <p>*((e5: +MEASURE_00 (x3)Theme (x1)Attribute) (e6: +BE_01 (x3)Theme (x6:+LENGTH_00)Attribute)(e7: +MEASURE_00 (x6)Theme (x7: 1 +METRE_00)Attribute))</p>

Tabla 3. El nuevo concepto \$SQUARE_METRE_00.

Sin embargo, en ocasiones, la unidad de medida que viene indicada en DBpedia no pertenece a un sistema internacional establecido (como es, p. ej., el sistema métrico decimal) y en su lugar se trata de otro concepto (p. ej. una ciudad, un año, etc.), por lo que también habría de comprobar que dicha unidad existe como concepto en la Ontología de FunGramKB antes de avanzar al siguiente paso. Esto significa que, además de las unidades de medida numéricas indicadas generalmente como “*label*”, podemos encontrar información adicional en la etiqueta “*range*” indicada en DBpedia, la cual puede aportar otros datos semánticos o conceptuales adicionales. Por tanto, en algunas reglas en las que el valor de la propiedad no es una unidad numérica, también podemos encontrar la necesidad de expresar algún tipo de indicación conceptual con respecto a su naturaleza, lo que en la sección 5.4.1.5 del próximo capítulo tratamos con más profundidad.

Por otro lado, también es importante contemplar que existen propiedades en las que no es necesario indicar información adicional con respecto a la naturaleza conceptual del valor de la propiedad, puesto que dicha inclusión podría resultar

redundante. Es el caso de propiedades como, por ejemplo, *address*, perteneciente a la clase “Building”:

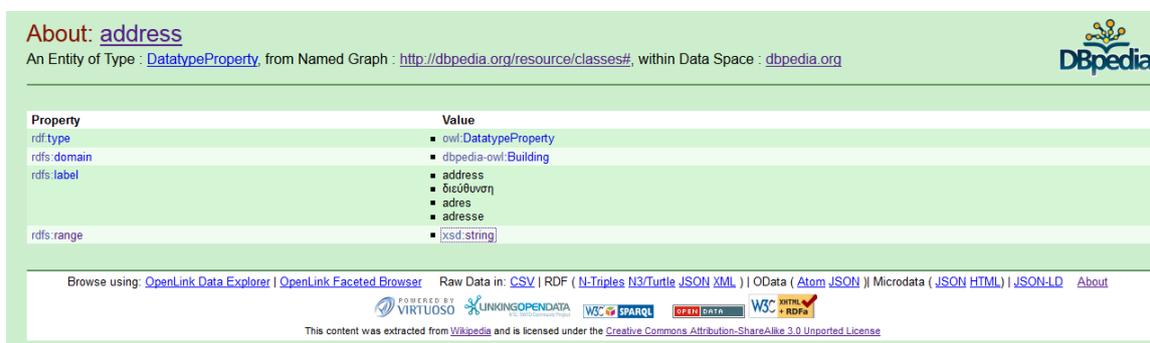


Ilustración 13. Propiedad *address* dentro de la clase “Building”⁷²

Propiedad	<i>address</i>
Clase	Building
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>))Theme (x2:+BUILDING_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+ADDRESS_00)Attribute (f1: <VAL>))Location)

Tabla 4. La propiedad *address* de la clase “Building” en el Onomasticón de FunGramKB.

El motivo por el cual se prescinde de indicar el valor de la propiedad es porque la información que la etiqueta “*label*” de la propiedad *address* indicada por DBpedia, como podemos ver en la Ilustración 13, ya es “*address*”, razón por la cual no consideramos necesario indicarlo de nuevo, con objeto de evitar la redundancia. En relación a la información proporcionada por la etiqueta “*range*”, DBpedia nos indica que será “*string*”, lo que se refiere a una secuencia o cadena de texto y que, por tanto, tampoco es necesario indicarlo al considerarse algo implícito, puesto que todos los valores no numéricos se expresan como texto.

⁷² <http://dbpedia.org/ontology/address>, fecha de consulta 23/03/2014.

Algo similar ocurre con la propiedad *nearestCity* de la clase “Place”, la cual hemos observado en la Ilustración 6 y cuya regla indicamos a continuación:

Propiedad	<i>nearestCity</i>
Clase	Place
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme (x1)Location (f1: m +NEAR_00)Position)

Tabla 5. La propiedad *nearestCity* de la clase “Place” en el Onomástico de FunGramKB.

En este caso, no es necesario indicar que el valor de la propiedad será un lugar poblado o “Populated Place”, que es lo que indica DBpedia, ya que dicho concepto en nuestra estructura COREL ya viene heredado. Esta herencia conceptual proviene del hecho de que tanto el concepto +TOWN_00 como el concepto +CITY_00 en la Ontología de FunGramKB ya recogen esta dimensión: la predicación (e4) en el postulado de significado de +TOWN_00 indica que “viven humanos”, algo que +CITY_00 hereda a través de la inclusión de +TOWN_00 en su definición, como podemos observar en (7) y (8):

(7)

CONCEPTO: +TOWN_00

POSTULADO DE SIGNIFICADO:

+(e1: +BE_00 (x1: +TOWN_00)Theme (x2: +ARTIFICIAL_AREA_00)Referent)

*(e2: +BE_01 (x1)Theme (x3: +BIG_00)Attribute)

*(e3: +COMPRISE_00 (x1)Theme (x4: m +BUILDING_00)Referent)

*(e4: +LIVE_01 (x5: +HUMAN_00)Theme (x1)Location)

(8)

CONCEPTO: +CITY_00

POSTULADO DE SIGNIFICADO:

+(e1: +BE_00 (x1: +CITY_00)Theme (x2: +TOWN_00)Referent)

*(e2: +BE_01 (x1)Theme (x3: m +BIG_00)Attribute)

Por tanto, una vez completada esta fase y determinado si es necesario o no indicar una unidad de medida o dimensión conceptual concreta en la regla que estamos creando, se debe proceder al cuarto y más complejo paso: la elaboración de la regla completa en lenguaje COREL.

4.3.1.4. Construcción de las predicaciones de la regla en lenguaje COREL

Con el fin de ilustrar de manera más clara la elaboración de la regla, continuaremos con el ejemplo de la propiedad *elevation*, perteneciente a la clase “Place”. Una regla es una plantilla en formato COREL que está configurada por una secuencia de predicaciones, las cuales sirven para expresar, al estilo de un postulado de significado, la relación existente entre una entidad y la clase a la que pertenece, una propiedad de esta entidad y los posibles valores que la entidad y la propiedad pueden adoptar. La entidad poseedora de la propiedad se representa por medio de la variable <C> (“concept”), mientras que colocaremos el valor de la propiedad en la variable <VAL> (“value”). Dentro del esquema COREL que compone la regla, se ha de indicar una primera predicación, denominada “e1”, que refleje la clase a la cual pertenece la propiedad cuya regla se está elaborando, algo a lo que nos referiremos como *contexto*. Mediante este contexto, conectamos los dominios de la Ontología con los del conocimiento del mundo presentes en el Onomasticón. En el caso de la propiedad *elevation*, la primera predicación se representa de la siguiente forma:

(9)

`+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent)`

El equivalente en lenguaje natural de la predicación (e1) podría expresarse como “*existe una entidad x1:<C> y esta entidad es siempre un lugar*”. El símbolo + que la predicación porta al frente indica que se trata de una predicación estricta, ya que una regla para propiedades que se encuentren dentro de la clase “Place” siempre requerirá que la entidad encajada en la posición <C> sea un lugar. La entidad concreta de la que se trate aparecerá de manera automática en la posición <C> una vez que se realice la proyección de datos desde DBpedia, una tarea que tendrá lugar tras la introducción de la regla en la herramienta DBpedia Mapper del Onomasticón.

A continuación, y una vez que se ha indicado en la predicación inicial (e1) la clase a la que pertenece la propiedad que estamos creando, se procederá a codificar el valor semántico de la propiedad. En el ejemplo que nos ocupa, se codifica de la siguiente forma:

(10)

`*(e2:+BE_01 (x1)Theme (x3: $ALTITUDE_00)Attribute (f1: <VAL>
+METRE_00)Quantity)`

El ejemplo (10) muestra la segunda predicación que compone la regla de *elevation*. En primer lugar, se observa que el símbolo * al frente de la predicación indica que ésta es rebatible, ya que el valor contenido en esta predicación (<VAL>) es susceptible de variar (por causas naturales, por ejemplo). En segundo lugar, se puede identificar que el concepto escogido para reflejar *elevation* se corresponde con el concepto terminal \$ALTITUDE_00. Gracias a los comentarios incluidos por DBpedia en la propiedad *elevation*, como se indicaba en el paso 2, es posible escoger el concepto

de la Ontología que mejor se corresponde con las características de la propiedad. La manera de buscar el concepto más apropiado coincide con el método de búsqueda de la unidad de medida que se ha descrito en el paso 3 y que se mostraba en la Ilustración 10 e Ilustración 11. En este caso, al introducir “elevation” como unidad léxica en la casilla de búsqueda, ésta no devuelve resultados, como observamos a continuación:

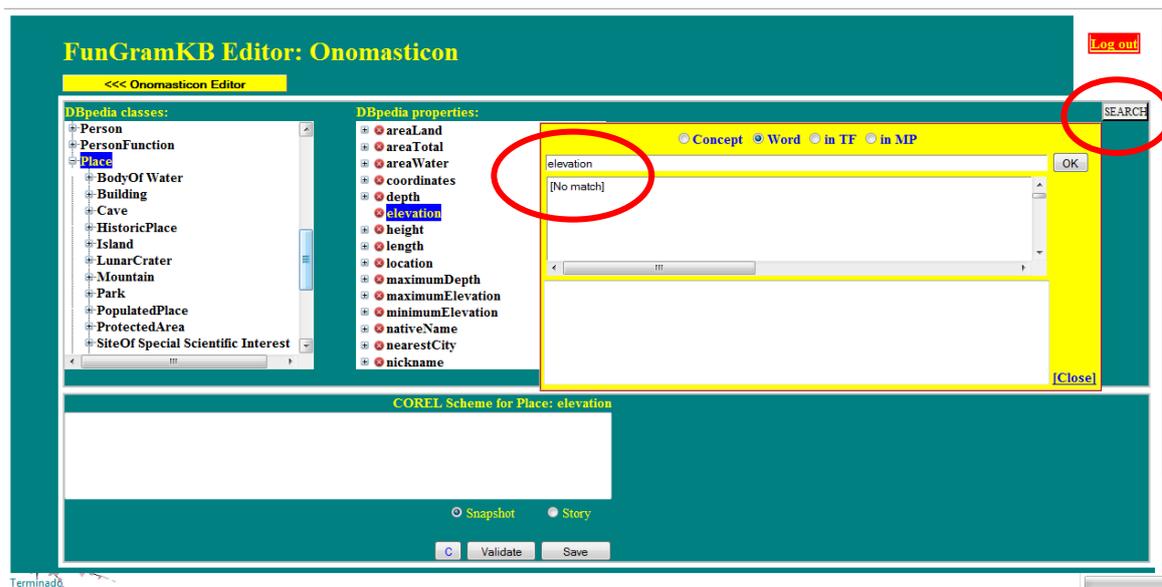


Ilustración 14. Búsqueda infructuosa de la unidad léxica “elevation” en *FunGramKB Editor* del Onomasticón.

A pesar de no haber obtenido resultado en la búsqueda de la unidad léxica “elevation”, sería poco prudente concluir sin mayor comprobación que el concepto que ésta designa no existe en la Ontología de FunGramKB. Recordemos que la Ontología de FunGramKB está estructurada en torno a conceptos y no en torno a unidades léxicas o palabras, de lo que se desprende que es posible que el concepto denotado por la propiedad *elevation*, descrita en los comentarios de DBpedia como “*average elevation above the sea level*” (como vemos en la Ilustración 7), sí exista en la Ontología de FunGramKB, pero lexicalizado de manera diferente. Con el propósito de averiguar este hecho, una opción es acudir a recursos lexicográficos que aporten una descripción extendida de la unidad léxica “elevation”, como por ejemplo la definición que se

obtiene tras la consulta del diccionario Longman Dictionary of Contemporary English Online⁷³, que define “elevation” como “*a height above the level of the sea*”⁷⁴. Al hallar la unidad léxica “height” dentro de la definición, podemos acometer una búsqueda de esta unidad en la Ontología de FunGramKB, gracias a lo cual se posibilita el encuentro con el concepto que mejor refleja la propiedad que se desea plasmar en la regla: el concepto \$ALTITUDE_00, como muestra la Ilustración 15:

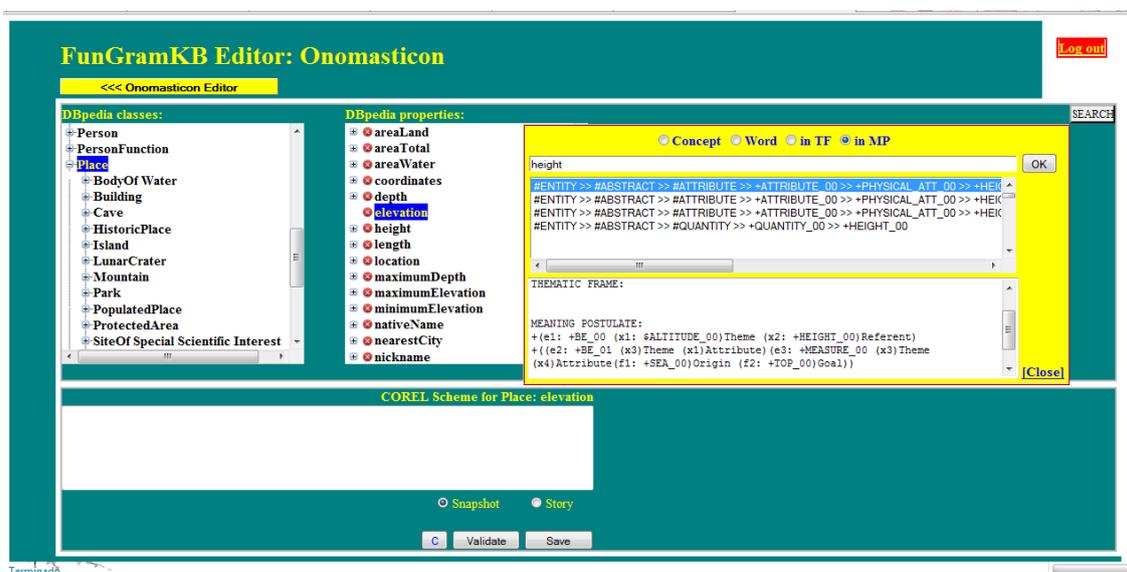


Ilustración 15. Identificación del concepto \$ALTITUDE_00 en FunGramKB a través de la lexicalización “height”.

Asimismo, a la hora de codificar una propiedad en lenguaje COREL, es muy importante velar por que no se produzca una confusión terminológica provocada por la posible similitud entre la denominación de un concepto en la Ontología de FunGramKB y la lexicalización en lenguaje natural que encontremos en la definición de DBpedia o de Schema.org y que pueda resultarnos similar a ella. Recordemos que la denominación en lenguaje COREL de los conceptos de la Ontología de FunGramKB puede no estar íntimamente relacionada con las lexicalizaciones de los mismos, por lo que siempre se recomienda verificar en la Ontología de FunGramKB tanto el postulado de significado

⁷³ <http://www.ldoceonline.com>

⁷⁴ <http://www.ldoceonline.com/dictionary/elevation>, fecha de consulta 06/12/2013.

de cada concepto como su descripción en lenguaje natural. Un ejemplo de ello lo encontramos en el concepto \$PUBLISH_00, el cual podría resultarnos a simple vista directamente relacionado con la propiedad *publisher* de la clase “Work” de DBpedia. Sin embargo, si comprobamos la definición en DBpedia y en Schema.org de esta propiedad, encontramos que en ambas fuentes la etiqueta “range” o valor de la propiedad es una empresa u organización, por lo que la propiedad *publisher* se debe interpretar entonces como “entidad u organización que lanza al mercado un producto creativo realizado por alguien”:

Property	Value
rdfs.type	owl:ObjectProperty
rdfs.domain	dbpedia-owl:Work
rdfs.label	publisher Herausgeber εκδότης uitgever
rdfs.range	dbpedia-owl:Company
owl:equivalentProperty	http://schema.org/publisher

Ilustración 16. La propiedad *publisher* en DBpedia⁷⁵

Property: **publisher**
The publisher of the creative work.

Values expected to be one of these types:
[Organization](#)

Used on these types:
[CreativeWork](#)

Schema Version 1.0f
[Terms and conditions](#)

Ilustración 17. La propiedad *publisher* en Schema.org⁷⁶

⁷⁵ <http://dbpedia.org/ontology/publisher>, fecha de consulta 18/03/2014.

⁷⁶ <http://schema.org/publisher>, fecha de consulta 18/03/2014.

Mientras, en FunGramKB, el concepto \$PUBLISH_00 se define del siguiente modo:

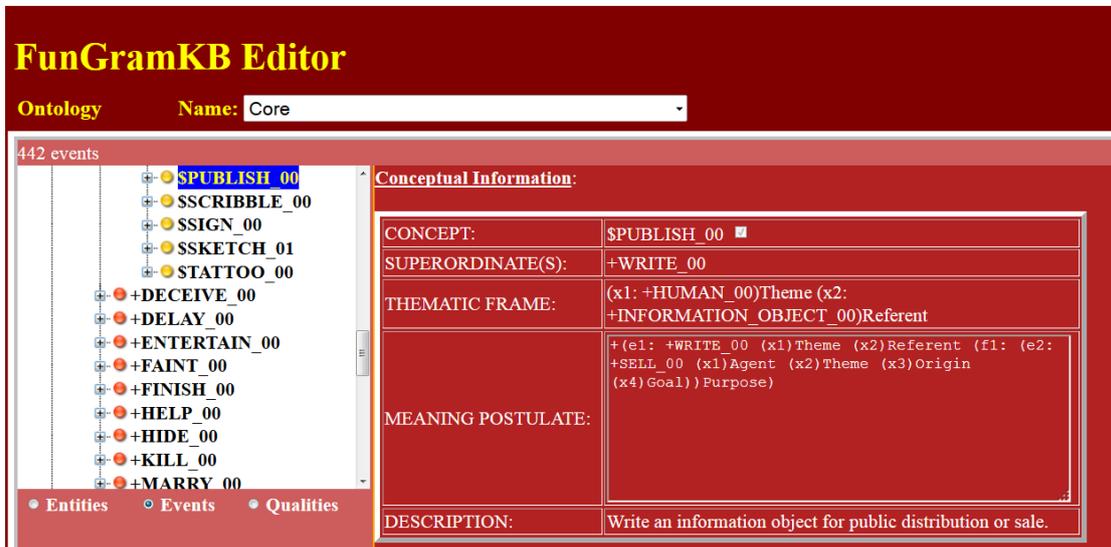


Ilustración 18. El concepto \$PUBLISH_00 en la Ontología de FunGramKB.

Por tanto, si para FunGramKB el concepto \$PUBLISH_00 se refiere a una persona que escribe algo para distribuirlo al público o venderlo, mientras que en DBpedia sabemos que esta propiedad alude a la empresa u organización que lanza al mercado dicha publicación, debemos ser capaces de modular esta información de manera adecuada en la regla que estamos creando, con objeto de modificar los valores conceptuales pertinentes para respetar el sentido original de la propiedad conferido por DBpedia. Puesto que, a pesar de estas diferencias, seguimos considerando que el concepto \$PUBLISH_00 es el que mejor encaja con la propiedad *publisher*, decidimos introducir un satélite de compañía para indicar que la acción de \$PUBLISH_00 no la realiza de manera unilateral el argumento denominado “x1”, sino con la ayuda de otro argumento, que no es otro que el elemento <VAL>, puesto que x1 es la entidad que escribe la obra, pero es la compañía quien lo lanza al mercado. Cabe especificar la importancia de no confundir el satélite “Company”, que denota el acompañamiento coordinado realizado por una entidad de manera conjunta con el argumento que lleva a

cabo (en este caso, x1) la ejecución de un evento, con el concepto +COMPANY_00 que denota a una empresa. Todo esto queda reflejado del siguiente modo en la regla elaborada en lenguaje COREL:

Propiedad	<i>publisher</i>
Clase	Work
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2: \$PUBLISH_00 (x3)Theme (x1)Referent (f1:<VAL>)Company)

Tabla 6. La propiedad *publisher* de la clase “Work”.

Así pues, una vez hallado el concepto más cercano al valor semántico de la propiedad cuya regla estamos creando, se procederá a insertarlo en la predicación de la manera más adecuada. No obstante, si bien en el caso del ejemplo *elevation* que hemos visto ha sido posible encontrar un concepto que se corresponde exactamente con el valor semántico de la propiedad (como observamos en la Ilustración 15), también puede ocurrir que, en otros casos, no sea posible hallar un concepto en la Ontología de FunGramKB que sea equivalente, de manera exacta, a este valor semántico. En estos casos, se deben tener en consideración diversos factores con objeto de tomar la decisión más adecuada para continuar con la elaboración de la regla. Para poder continuar con la creación de la regla se puede decidir o bien crear un nuevo concepto terminal en la Ontología de FunGramKB –como se ha hecho con la unidad de medida m² mostrada anteriormente– el cual refleje de manera exacta el valor semántico que necesitamos, o bien describir el nuevo concepto mediante el empleo de otros ya existentes en la Ontología, los cuales permitan la construcción conceptual del valor semántico que

necesitamos reflejar⁷⁷. La decisión, por tanto, de crear un nuevo concepto o de reflejar el mismo mediante la combinación de otros ya existentes debe realizarse tras un proceso de negociación, en el que es necesario sopesar tanto la relevancia de la creación del nuevo concepto como la conveniencia de que éste posea una granularidad semántica muy fina. También podemos abogar por una posición intermedia dentro de este debate, consistente en la creación de un subconcepto dentro de un concepto ya existente, lo que no modifica la jerarquía conceptual ni añade nuevos conceptos sino que, simplemente, modifica o concreta la naturaleza de los participantes en el evento reflejados en el Marco Temático del concepto. Un subconcepto se define de la siguiente manera:

“Subconcepts are not deemed to be real conceptual units but lexically-motivated refinements of the selectional preferences in the thematic frame of already existing basic or terminal concepts, which serve as superconcepts. Subconcepts share all conceptual properties of their superconcepts, except for some of the selectional preferences in their thematic frames: even the number and role of participants in those thematic frames must be identical.”

(Periñán Pascual y Mairal Usón, 2011:29).

Esto es, un subconcepto vinculado a un concepto tan sólo modifica las preferencias de selección de éste, de tal forma que los postulados de significado siguen siendo los mismos, aunque los participantes del evento se modifiquen. Por ejemplo, el concepto +SOUND_00 se define como “*Utter a sound through the mouth*” y el participante prototípico del evento es un animal, expresado en el Marco Temático de este modo:

MARCO TEMÁTICO:	(x1: +ANIMAL_00)Theme (x2: +SOUND_00)Referent
-----------------	---

Sin embargo, dentro de este concepto encontramos el subconcepto –BARK, que se define de igual forma que +SOUND_00 en términos de su postulado de significado,

⁷⁷ Esta situación no debe ser considerada una desventaja, sino todo lo contrario, con objeto de tener una representación de significado más rica dentro del Onomasticón.

pero que tiene como participante principal del evento al concepto +DOG_00, expresado así:

MARCO TEMÁTICO:	(x1: +DOG_00)Theme (x2: +SOUND_00)Referent
-----------------	--

Un caso similar ocurre con el subconcepto –BLEET, que tiene por participantes +GOAT_00 o +SHEEP_00:

MARCO TEMÁTICO:	(x1: +GOAT_00 ^ +SHEEP_00)Theme (x2: +SOUND_00)Referent
-----------------	---

No obstante, en otras ocasiones se toma la decisión de crear un nuevo concepto completo por analogía con conceptos similares o paralelos ya existentes en la Ontología de FunGramKB. Es el caso del concepto \$ARCHIPELAGO_00, cuya creación se ha estimado necesaria para la elaboración de la propiedad *archipelago* de la clase “Island”, que se encuentra bajo el concepto padre “Place”. La motivación para crear este nuevo concepto se apoya, en primer lugar, en la necesidad de contar con el mismo tras comprobar su ausencia en la Ontología de FunGramKB, ya que si la población de ésta se encontrara completamente terminada, podríamos utilizar cualquier concepto que pudiéramos necesitar. Para crear el nuevo concepto \$ARCHIPELAGO_00, nos apoyamos en la existencia de diversos conceptos integrados en la Ontología de FunGramKB en la jerarquía del concepto básico +GROUP_00, los cuales se utilizan para aludir a diversos constructos conceptuales que denotan una agrupación de entidades iguales, como por ejemplo \$CUTLERY_00 o \$SYNDICATE_00, entre otros muchos que podemos ver a continuación:

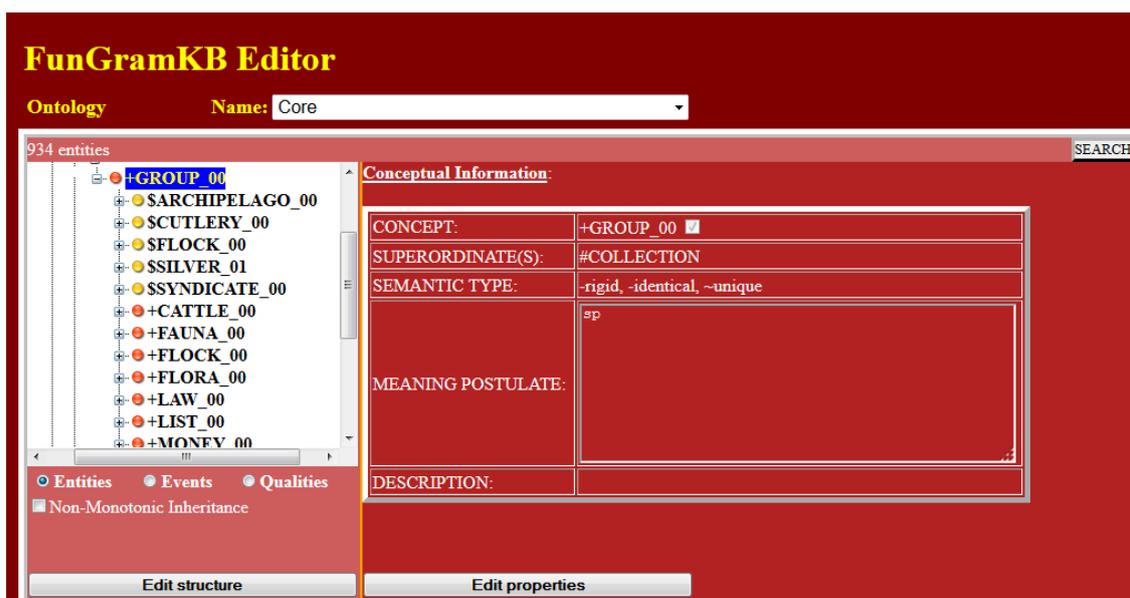


Ilustración 19. Jerarquía conceptual bajo el concepto +GROUP_00.

Otro caso similar es el del concepto \$PARTY_01, referido a un partido político, el cual no existía en la Ontología de FunGramKB, motivo por el cual se ha creado dicho concepto al amparo de +ORGANIZATION_00, hiperónimo bajo el que encontramos el nuevo concepto \$PARTY_01:

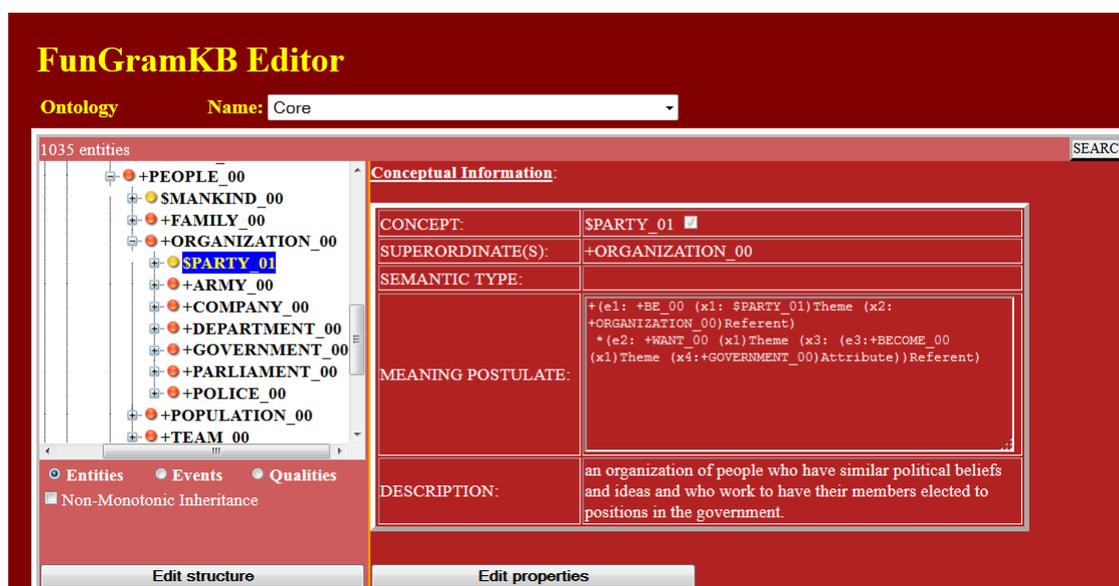


Ilustración 20. Jerarquía conceptual en la que se encuadra el nuevo concepto \$PARTY_01.

Retornando a la predicación (e2) que aparece en el ejemplo (10), se advierte también la presencia del concepto +METRE_00, que indica la unidad de medida en la

que se expresa el valor <VAL>. Es en este momento cuando detectamos la necesidad de indicar la unidad de medida, como ya se había anticipado en el paso 3 de la metodología. Si se realiza una equivalencia en lenguaje natural de lo que la predicación (e2) refleja en lenguaje COREL, podría expresarse como: “*la entidad (x1)*⁷⁸ *posee una propiedad que es la altitud y que obtiene el valor de <VAL> metros*”. Puesto que esta predicación es rebatible, ya que la altitud de un lugar podría cambiar a lo largo del tiempo, nótese que no se indica ningún adverbio absoluto como “siempre” o “nunca” en la equivalencia en lenguaje natural del esquema COREL. Mediante la combinación de ambas predicaciones, (e1) y (e2), se obtiene la regla final, que para la propiedad *elevation* de la clase “Place” es la siguiente:

(11)

+ (e1:+BE_00 (x1:<C>)Theme (x2:+PLACE_00)Referent)

* (e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute

(f1:<VAL>+METRE_00)Quantity)

Tras la elaboración de la regla, procedemos a introducirla en el correspondiente recuadro de “COREL scheme for Place: elevation” dentro del FunGramKB Editor del Onomasticón, así como a realizar las pertinentes operaciones de validación sintáctica de la regla, una posibilidad que ofrece la aplicación mediante la opción “*Validate*” presente bajo la zona de edición, como se puede observar en la Ilustración 21:

⁷⁸ La entidad (x1) aparece de nuevo y por co-indización se refiere a la misma entidad (x1:<C>) de la anterior predicación, (e1), que ya se ha indicado que es siempre un lugar.

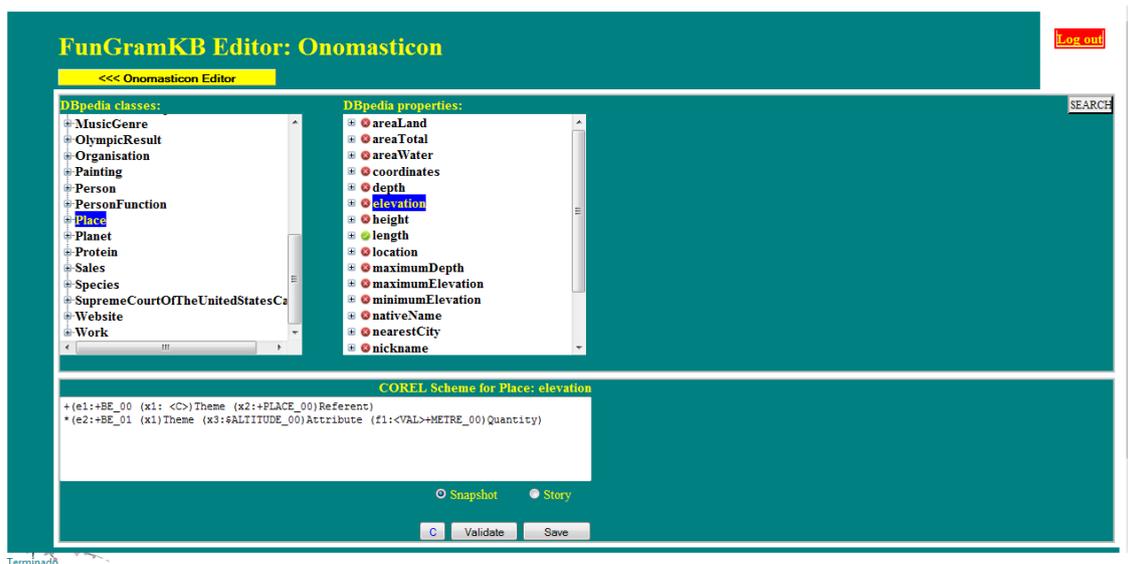


Ilustración 21. Introducción de la regla para la propiedad *elevation* en FunGramKB Editor del Onomasticón.

Después de presionar la opción “Save”, ubicada a la derecha de la opción de validación, como se muestra en la imagen anterior, la regla que refleja la propiedad *elevation* habrá quedado almacenada finalmente en el Onomasticón de FunGramKB.

4.3.2. LA FASE DE PROYECCIÓN

Con posterioridad a la fase de construcción que se ha descrito en la sección anterior, la cual culmina con la introducción de cada regla en el Onomasticón de FunGramKB, tiene lugar la denominada fase de proyección. Esta fase consiste en la implementación del proceso computacional que permite la proyección de los datos de fuentes externas (DBpedia) en las reglas introducidas en el Onomasticón, de forma que se completan con los datos obtenidos de dichas fuentes. Es decir, las variables <C> (*Concept*) y <VAL> (*Value*) de cada una de las reglas son sustituidas por la información contenida en las fichas informativas de Wikipedia y obtenida a través de DBpedia. De este modo, las plantillas del Onomasticón se instancian para convertirse en predicaciones que configuran los retratos o historias correspondientes en cada propiedad de cada entidad nombrada.

Para ilustrar el producto de la fase de proyección, podemos tomar algunos de los ejemplos de propiedades mostradas en la anterior fase de construcción. Mediante la regla de la propiedad *elevation* pueden obtenerse los siguientes esquemas conceptuales:

(12)

a)

$+(e1:+BE_00 (x1: \%TOLEDO_SPAIN_00)Theme (x2:+PLACE_00)Referent)$
 $*(e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute (f1:529$
 $+METRE_00)Quantity)$

Toledo (España) es un lugar ubicado a 529 m de altitud sobre el nivel del mar.

b)

$+(e1:+BE_00 (x1: \%SAN_FRANCISCO_00)Theme (x2:+PLACE_00)Referent)$
 $*(e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute (f1:15.8496 +METRE_00)$
 $Quantity)$

San Francisco es un lugar ubicado a 15,8496 m de altitud sobre el nivel del mar.

c)

$+(e1:+BE_00 (x1: \%CASTELLON_DE_LA_PLANA_00)Theme$
 $(x2:+PLACE_00)Referent)$
 $*(e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute (f1: 30 +METRE_00)$
 $Quantity)$

Castellón de la Plana es un lugar ubicado a 30 m de altitud sobre el nivel del mar.

Como se desprende de los ejemplos en (12), la proyección realizada mediante el proceso computacional materializa la sustitución de la variable <C> por un concepto que representa a una entidad nombrada concreta, así como la del valor <VAL> por otro concepto o por un operador de cuantificación (p. ej. un número), según corresponda, produciendo de este modo una identificación entre las entidades nombradas, sus propiedades y el valor real que éstas poseen, según los datos contenidos en las fichas informativas de Wikipedia, proyectados a través de DBpedia en el Onomasticón de FunGramKB.

Otro ejemplo que podemos mostrar es el de la propiedad *publisher* que hemos visto en secciones anteriores, en el cual aprovechamos para ilustrar la secuencia completa de proyección a través de la figura que mostramos a continuación:



Figura 10. Proyección de la instancia “The Da Vinci Code” en la regla para la propiedad *publisher*.

En la Figura 10 podemos observar cómo se efectúa la proyección del valor de la propiedad *publisher* obtenido por una instancia concreta de la clase “Work”. En primer lugar, en la imagen 1 observamos la ficha informativa tal como se muestra en el artículo en Wikipedia de la entidad “The Da Vinci Code”⁷⁹. A continuación, podemos observar en 2 una imagen del código fuente a través del cual se materializa la ficha

⁷⁹http://en.wikipedia.org/wiki/The_DaVinci_Code fecha de consulta 26/03/2014.

informativa⁸⁰ en Wikipedia, el cual sirve de origen de datos para que FunGramKB importe dicha información en la plantilla COREL que refleje la propiedad *publisher*, obteniendo la proyección que observamos en 3 y que puede traducirse en lenguaje natural como “*The Da Vinci Code es una obra que es publicada por Doubleday (US), Transworld y Bantam Books (UK)*”.

4.3.3. LA FASE DE ACTUALIZACIÓN

Una vez que se han llevado a cabo las fases de construcción y de proyección, el trabajo del Onomasticón no se termina. En la fase subsiguiente a la construcción y proyección, llamada *fase de actualización*, tiene lugar el mantenimiento de la información incluida en FunGramKB y obtenida de las fuentes, principalmente de DBpedia. Se trata, pues, de una cadena de actualización informativa, mediante la cual posibilitamos que la ingente cantidad de información contenida en grandes repositorios de información en permanente cambio, como son Wikipedia y DBpedia, sea trasvasada a FunGramKB.

No obstante, esta fase de actualización forma parte de las futuras investigaciones dentro de FunGramKB, ya que dentro de dicha fase no sólo debe existir la posibilidad de actualizar el conocimiento nuevo procedente de DBpedia, sino también de actualizar los retratos e historias del Onomasticón con el conocimiento de otras fuentes.

En el presente capítulo hemos cubierto exhaustivamente el primero de los objetivos de esta tesis doctoral, es decir, *detallar una metodología de población semi-automática del módulo Onomasticón de FunGramKB*. Para ello, hemos enumerado de manera pormenorizada los pasos que componen la metodología de creación de reglas en

⁸⁰ http://en.wikipedia.org/w/index.php?title=The_Da_Vinci_Code&action=edit fecha de consulta 26/03/2014.

el Onomasticón de FunGramKB, la cual se compone, en primer lugar, de una fase de construcción, que es la más laboriosa, al tratarse de la parte manual del trabajo, si bien es asistida por las herramientas que FunGramKB pone a disposición del usuario (i.e. FunGramKB Suite, DBpedia Mapper, etc.). Esta fase comprende varios pasos, a lo largo de los cuales pueden surgir diversas necesidades que precisan atención (por ejemplo, la necesidad de crear un concepto nuevo en la Ontología). A continuación, se ha indicado que tienen lugar las dos fases automáticas del proceso: la fase de proyección, que posibilita el trasvase de la información de las fuentes a FunGramKB y la fase de actualización, que mantendrá dicha información actualizada.

En el siguiente capítulo presentamos los resultados de la aplicación de la metodología mostrada en el actual capítulo, es decir, las reglas creadas en lenguaje COREL y almacenadas en el Onomasticón, junto a las diversas consideraciones y peculiaridades observadas durante la elaboración de las mismas y los varios tipos de estructuras y construcciones que las componen. Esta tarea nos lleva a cubrir el segundo objetivo de esta tesis doctoral, puesto que efectuamos un *análisis y clasificación de las reglas* atendiendo a diversos criterios (semántico y formal), junto a las conclusiones que de ello se puedan derivar, con el fin de *confirmar o rebatir la hipótesis de que una estructura formal similar en los constructos elaborados puede indicar una motivación semántica compartida*.

CAPÍTULO V

ANÁLISIS DE RESULTADOS

5.1. PRESENTACIÓN

En el capítulo anterior se ha mostrado con detalle la metodología para la elaboración en lenguaje COREL de las plantillas denominadas *reglas*, mediante las cuales se posibilita la importación a FunGramKB de una gran cantidad de conocimiento cultural, procedente de fuentes que se encuentran en permanente actualización a través de la herramienta DBpedia.

Para la presente tesis doctoral se ha trabajado con un total de 231 propiedades de DBpedia que nos permitirán construir esquemas conceptuales que formarán parte del Onomasticón de FunGramKB. Estas reglas se incluyen en el Apéndice I, presentando así los resultados de la ejecución del proceso detallado en el Capítulo IV para la creación asistida de las mismas. No obstante, a pesar de ser su finalidad primordial, la creación de estas reglas no encuentra su único uso en permitir la importación de datos de la base DBpedia al Onomasticón de FunGramKB. Existe, asimismo, una finalidad de corte más lingüístico que se suma al propósito computacional de la creación de reglas, lo que ya se anticipaba en Carrión Varela (2010) como una hipótesis que pretendemos confirmar o rebatir en el presente trabajo y que forma parte del segundo de los objetivos

de esta tesis doctoral. Esta hipótesis parte del siguiente hecho: si se observa con detalle cada una de las reglas, una característica notable es que muchas de ellas presentan una estructura análoga, por lo que pueden agruparse de manera similar. Esta taxonomía de reglas serviría para facilitar el proceso de creación de éstas, ya que su estructuración interna puede ser prevista dependiendo del tipo de regla que se quiera construir. Esta característica puede favorecer futuras implementaciones para automatizar un poco más el proceso de proyección, es decir, para transitar a un proceso más *semi-automático*.

En Carrión Varela (2010) se presentaban las reglas elaboradas para 17 propiedades descriptivas de la clase genérica “Place” (sin subclases) del Onomasticón de FunGramKB, las cuales eran catalogadas conforme a criterios basados en observaciones formales de las reglas correspondientes a las 17 propiedades, lo que arrojaba una clasificación en cuatro grandes grupos:

1) Propiedades geográficas simples que codifican una dimensión:

Se trata de propiedades como *elevation*, *height*, *width*, *areaTotal*, *depth*, *length* y *location*, las cuales parecen reflejar propiedades promedio y no valores absolutos.

2) Propiedades en las que interviene una especificación de tipo superlativo:

Estas reglas ilustran aquellas propiedades en las que interviene una especificación de tipo superlativo, mediante un pre-modificador como “*maximum*” o “*minimum*” (*maximumElevation*, *minimumElevation*, *maximumDepth*) o un superlativo, como en *nearestCity*.

3) Propiedades complejas que precisan la introducción de otros conceptos que reflejen las circunstancias de ubicación espacial:

Se trata de otro tipo de propiedades más complejas, que demandan la integración de otros conceptos en el esquema, como muestran las reglas correspondientes a las propiedades *areaWater*, *areaLand* y *percentageOfAreaWater*.

4) Propiedades para las que no es preciso elaborar una regla atendiendo a diversa motivación:

Se encuentran en esta categoría las propiedades *otherName*, *nickname* y *nativeName*, para las cuales no se estima necesaria la creación de una regla, tanto por motivos técnicos de la herramienta como con objeto de minimizar la redundancia, ya que aluden a formas alternativas de nombrar la entidad. Esto puede ser resuelto mediante un proceso computacional que permita una redirección a la entidad principal y que muestre denominaciones alternativas de la misma en una sección que se designará como “*alternative name box*”. Un ejemplo de esto podría observarse en una entidad como la Mona Lisa (%MONA_LISA_00), a la cual se redireccionarían consultas que contuvieran sobrenombres como “La Gioconda” o “La Joconde”. Otra propiedad de la que se ha decidido prescindir a la hora de elaborar el inventario de reglas es *type*, al resultar redundante, puesto que esta propiedad ya queda reflejada en la jerarquía en la que se encuadre la entidad (i.e. la clase y la subclase a la que pertenezca). En última instancia, se observa la propiedad *coordinates*, para la cual no se ha elaborado regla, ya que no se estima relevante para el PLN la codificación de propiedades como coordenadas de longitud y latitud. No obstante, no se ha de descartar que, en el futuro, esta propiedad u otras que han sido inicialmente descartadas de manera

argumentada, adquieran un estatus de mayor relevancia debido a futuras aplicaciones de FunGramKB y, por tanto, también sean tratadas por medio de reglas en el Onomasticón.

Tras la descripción arriba detallada de estos grupos y las razones esgrimidas para clasificar las reglas en cada uno de ellos, en Carrión Varela (2010) se enunciaba la hipótesis que reproducíamos en el Capítulo I del presente trabajo como parte del segundo objetivo de esta tesis doctoral: una similitud formal o conceptual en la estructura de las reglas puede ser el reflejo de una motivación semántica compartida entre los constructos conceptuales que éstas representan.

Por tanto, con el objetivo de verificar si, efectivamente, la clasificación original de las reglas de la clase “Place” esbozada en Carrión Varela (2010) mantiene su validez a la luz de la creación de un número mucho más amplio de reglas en diversas clases y subclases, en el presente capítulo realizamos, en primer lugar, una descripción general de las reglas elaboradas para el actual trabajo. Posteriormente, presentamos un análisis crítico de la clasificación original, lo que incluye el ejercicio de evaluar si es posible o no mantener dicha clasificación. Finalmente, mostramos otras alternativas taxonómicas, tanto semánticas como formales, que estimamos más adecuadas al estado actual de elaboración de reglas dentro del Onomasticón de FunGramKB.

5.2. REGLAS ELABORADAS PARA EL PRESENTE TRABAJO

Al amparo del actual trabajo se ha trabajado en un total de 231 propiedades de DBpedia que permitirán la proyección de los contenidos de este recurso hacia el Onomasticón de FunGramKB. Además de las 17 propiedades pertenecientes a la clase genérica “Place”, a este número se añaden 214 propiedades de las diversas subclases de “Place”, completando así este dominio en su totalidad, además de reglas pertenecientes

a otras clases, que son “Person”, “Work” y “Organisation”. Se han elegido estas clases y no otras para proseguir con la población del Onomasticón de FunGramKB por tratarse de las categorías más representativas dentro de Wikipedia y del mundo en general, ya que en el inventario de entidades nombradas que DBpedia ha clasificado, las categorías con más frecuencia utilizadas son los topónimos, los nombres propios de personas, las obras de cualquier tipo (p. ej. literarias, musicales, cinematográficas, digitales, etc.), así como cualquier clase de organización (p. ej. empresas, organismos, estamentos gubernamentales, agencias, entidades educativas, etc.), según observamos en la siguiente tabla:

Class	Instances
Resource (overall)	3,220,000
Place	639,000
Person	832,000
Work	372,000
Species	226,000
Organisation	209,000

Tabla 7. Número de entidades dentro de las clases más pobladas en DBpedia⁸¹

Podemos observar que entre estas categorías tan pobladas se encuentra la clase “Species”, que hace alusión a cualquier tipo de especie animal o vegetal clasificada dentro de una taxonomía científica. Debido a su extremo carácter especializado y a la poca productividad ontológica que encontramos que aportaría a una base de conocimiento de PLN genérica como es FunGramKB, no hemos considerado su inclusión en este momento en la población del Onomasticón. No obstante, dicha clase puede resultar de gran utilidad para la elaboración de una ontología especializada si en futuros trabajos se decidiera incorporar en FunGramKB una ontología satélite de

⁸¹ Tabla obtenida de <http://wiki.dbpedia.org/ontology>, fecha de consulta 24/03/2014.

términos científicos, al igual que ya cuenta con una subontología de terminología jurídica y legal, como hemos visto en el Capítulo III.

5.3. ANÁLISIS DE LA CLASIFICACIÓN ORIGINAL

Si observamos con detenimiento la clasificación de las reglas de la clase “Place” efectuada en Carrión Varela (2010), podemos detectar un rasgo clave que parece funcionar como elemento motivador de la misma: la elevada similitud formal de las reglas incluidas en ella. A continuación, amparándonos en dicho razonamiento, enumeramos una serie de consideraciones sobre esta clasificación inicial.

En primer lugar, con respecto a las reglas creadas para la clase “Place”, observamos que a la hora de elaborar tanto la primera predicación de cada regla, la cual denominamos (e1) y que vincula a una entidad con una clase, como a la hora de elaborar el resto de predicaciones, resulta relativamente sencillo efectuar este vínculo, puesto que la lista de propiedades parece estar compuesta en su gran mayoría de conceptos básicos, entendido esto tanto en el sentido genérico de la palabra (i.e. conceptos de uso cotidiano) como en el sentido estricto de la clasificación conceptual de FunGramKB (i.e. conceptos *básicos*, precedidos por el símbolo +). Así pues, a pesar de que en 7 de las 17 reglas (*areaLand*, *areaTotal*, *areaWater*, *elevation*, *maximumElevation*, *minimumElevation* y *percentageOfAreaWater*) aparecen conceptos terminales, encontramos que su aparición está principalmente vinculada a la unidad de medida (p. ej. \$SQUARE_METRE_00 y \$PERCENTAGE_00) y no a la propiedad en sí, a excepción de las propiedades que aluden a “elevation”, en las cuales se utiliza el concepto \$ALTITUDE_00. No obstante, no quiere esto decir que las reglas sean simples en su totalidad, sino que parecen seguir un patrón homogéneo, a saber: la primera predicación (e1) vincula a la entidad con la clase, mientras que la segunda y, si

la hubiera, la tercera, efectúan la codificación de la propiedad, quedando siempre la inserción del elemento <VAL> en la última predicación, ya sea ésta (e2) o (e3).

Si efectuamos un análisis de los valores semánticos agrupados en cada categoría de la tipología descrita, observamos que las diferentes categorías aglutinan respectivamente:

- 1) Propiedades que designan cualidades simples de una entidad (concretamente, dimensiones físicas o espaciales).
- 2) Propiedades que introducen un pre-modificador (i.e. máximo/mínimo).
- 3) Propiedades que aglutinan más de un concepto.
- 4) Propiedades para las que no se estima necesaria la creación de una regla.

La primera categoría, que comprende cualidades simples de una entidad, presenta una estructura en COREL en la que la predicación (e2), que aparece en segundo lugar, adopta una estructura de papeles temáticos del tipo “Theme + Attribute + Quantity”. Es decir, que una entidad (*Theme*) posee una propiedad (*Attribute*) que se cuantifica con un valor numérico (*Quantity*). Podemos concluir, por tanto, que las propiedades que se encuadran en esta categoría se podrían denominar *propiedades atributivas*, puesto que éstas se asocian de manera intrínseca a la naturaleza de la entidad de la que estamos hablando. En otras palabras, no se trata de una vinculación externa con otras entidades, sino de un vínculo entre una entidad y un atributo, el cual adquiere un valor determinado cuantificable para dicha entidad. Las denominadas

propiedades atributivas que compartirían este rasgo semántico son *elevation*, *height*, *width*, *areaTotal*, *depth*, *length* y *location*.

En segundo lugar, encontramos una serie de propiedades que, si bien contienen también atributos y una cuantificación del valor de los mismos, lo hacen con un pre-modificador que codifica valores absolutos (*maximum/minimum*) y no valores promedio, como sucedía en la anterior categoría. En este caso, la estructura formal de las reglas continúa mostrando dos predicaciones, cuya segunda predicación mantiene el esquema “Theme + Attribute + Quantity” de la categoría anterior, si bien aparece un satélite, (f2), en la segunda predicación, que añade el papel temático de “Scene” o “Position” con sus respectivos papeles dentro del satélite, cuya función es precisar que esa dimensión tiene lugar en un momento y/o punto geográfico determinados, en contraposición a las *propiedades atributivas* simples que abogan por reflejar un valor promedio de la dicha dimensión. Para denominar a las reglas de esta categoría, podemos utilizar el término *propiedades atributivas absolutas*, dado que reflejan el valor más extremo que puede presentar un rasgo atributivo de una entidad.

En tercer lugar, encontramos una nueva clase de propiedades en las que no se codifica únicamente una entidad y un rasgo o atributo de la misma, sino que entran en juego elementos adicionales. Esto es, para poder codificar la dimensión que estas propiedades reflejan, necesitamos utilizar entidades adicionales a la entidad <C>, las cuales, en última instancia, son las que obtienen el valor cuantificado en <VAL>. Es el caso de propiedades como *areaWater*, *areaLand*, *percentageOfAreaWater* y *nearestCity*, en las que la entidad o concepto <C> no se encuentra solo en el constructo conceptual que estamos elaborando, sino que precisa de la interrelación con otras entidades para poder ser comprendido. Por ejemplo, los esquemas de estas propiedades precisan la introducción de otros conceptos que reflejen las circunstancias de ubicación

espacial, lo cual se manifiesta a través de los satélites de posición, “Position”, que aparecen en las predicaciones. Esta inclusión de nuevos elementos provoca que el número de predicaciones tienda a aumentar como norma general a tres en lugar de mantenerse en dos. Este planteamiento deriva en la aserción, basada en los resultados obtenidos hasta ahora, de que, mientras la primera predicación siempre indica la clase a la que pertenece la regla y la entidad que posteriormente se proyecte en ella (i.e. la variable <C>), la última predicación es siempre la que contiene el valor que la propiedad puede adoptar (i.e. la variable <VAL>), sin perjuicio de que sea preciso intercalar otras predicaciones entre la primera y la última con objeto de completar la definición de la propiedad que la regla denota (p. ej., una predicación que contextualice espacialmente la propiedad codificada). Asimismo, apreciamos que a esta categoría se adscribe la propiedad *nearestCity*, la cual previamente había sido considerada similar a las reglas que contienen un premodificador (máximo/mínimo) y que hemos denominado propiedades atributivas absolutas. El motivo de este cambio reside en el hecho de que, a pesar de contener un superlativo, en esta propiedad intervienen entidades adicionales (i.e. *city*), razón por la cual consideramos que esta propiedad debe ser encuadrada más correctamente en la actual categoría y no en la anterior.

Para ilustrar más claramente esta situación, podemos observar el ejemplo de la propiedad *areaWater*. Con objeto de poder reflejar esta propiedad, es necesario contextualizar, mediante la inclusión de conceptos adicionales, que la entidad <C> es un lugar dentro del cual existe un área de agua, la cual tiene unas dimensiones de <VAL> m². A diferencia de las propiedades denominadas *atributivas*, el valor que se refleja en <VAL> no se atribuye de manera intrínseca a la entidad <C>, sino a otra entidad vinculada estrechamente con <C>. Por tanto, no basta con asignar el espacio <C> a la entidad de la que estamos hablando, sino que es preciso añadir otras entidades dentro de

la regla que estamos elaborando para que ésta pueda realizarse de manera completa. Por ende, puesto que las propiedades reflejadas no parecen ser atributos intrínsecos a la naturaleza de la entidad, sino que más bien se trata de valores que obtiene otra entidad relacionada con <C>, no parece adecuado mantener la denominación de *atributivas* para referirnos a este tipo de propiedades. Por contra, a este tipo de propiedades se les podría denominar *propiedades predicativas*, ya que no basan su estructura únicamente en el esquema “Theme +Attribute + Quantity” o equivalente, sino que precisan de una estructura predicativa en la que otros elementos adicionales entran en juego (i.e. nuevas entidades, nuevos eventos), los cuales contextualizan tanto a la entidad <C> como a las demás entidades vinculadas a las que se atribuye el valor de la variable <VAL>. Asimismo, podemos incluso efectuar una subdivisión dentro de la categoría *predicativa*, con objeto de separar aquellas propiedades en cuyas predicaciones interactúa más de una entidad, i.e. *propiedades predicativas relacionales* (p. ej. *areaWater*) y aquellas cuyas predicaciones hacen única alusión a la entidad <C>, siendo entonces un tipo de *propiedades predicativas intrínsecas*⁸².

Por tanto, si deseamos proponer una nueva nomenclatura basada en las reglas elaboradas para la clase “Place” y descritas en Carrión Varela (2010), ésta podría contener las cuatro categorías siguientes: *propiedades atributivas simples*, *propiedades atributivas absolutas*, *propiedades predicativas relacionales* y *propiedades predicativas intrínsecas*.

Finalmente, la cuarta de las categorías originales y la quinta en esta nueva clasificación es aquella en la cual se engloban las propiedades para las que no se estima necesaria la creación de una regla. Esta categoría podría ser subdividida, a su vez, en

⁸² No encontramos en “Place” ningún ejemplo de este tipo de propiedades, pero sí en otras clases del Onomasticón que mencionamos más adelante.

dos sub-categorías acordes con la nomenclatura utilizada para las mismas en el Onomasticón de FunGramKB, a saber: aquellas propiedades que constituyen nombres alternativos de una entidad tendrán asignadas el código “a/n” (*alternative name*) y aquellas entidades cuya regla se ha decidido no elaborar por otras razones tendrán el código “n/r” (*no rule*). En el Apéndice I se pueden identificar las reglas pertenecientes a cada una de estas dos sub-categorías.

Una vez elaborada esta nueva nomenclatura, cabe reformularse la pregunta acerca de si la misma es producto de una observación formal de las reglas elaboradas o si, por el contrario, se trata de una taxonomía creada al amparo de un análisis semántico de las mismas. Como ya anticipábamos al principio de este trabajo, una similitud formal en las reglas puede llevarnos a descubrir una similitud semántica, por lo que, en ocasiones, puede resultar harto complicado determinar qué lleva a qué, si una observación formal a una apreciación semántica o viceversa. En este caso, quizás podamos afirmar que, si bien la estructura formal es la que primero ha aportado indicios acerca de la similitud entre reglas, se trata de la semántica propia de cada una de ellas la que finalmente ha propiciado la división atributiva/predicativa, ya que se trata de una clasificación basada en las relaciones que se establecen entre la entidad <C> y el valor <VAL>. Por tanto, con objeto de reafirmar esta clasificación, la cual resumimos en la Tabla 8, como más semántica que formal, creemos necesario elaborar una clasificación adicional bajo nuevos criterios, esta vez estrictamente formales, con el fin de catalogar de esa manera las reglas elaboradas.

Tipos de propiedades ATRIBUTIVAS	
Simples Ejemplo: Place/ <i>width</i>	Absolutas Ejemplo: Place/ <i>maximumElevation</i>
Tipos de propiedades PREDICATIVAS	
Relacionales Ejemplo: Place/ <i>areaWater</i>	Intrínsecas Ejemplo: Person/ <i>birthPlace</i>

Tabla 8. Tipos de propiedades atendiendo a criterios de clasificación semánticos.

Por tanto, en la siguiente sección presentamos una nueva alternativa taxonómica para clasificar las reglas del Onomástico de FunGramKB, basada en criterios más formales que semánticos y a la luz de la diversidad de estructuras que han surgido durante la elaboración de las reglas para las 231 propiedades en las que hemos trabajado.

5.4. ALTERNATIVAS TAXONÓMICAS PARA CLASIFICAR LAS REGLAS ELABORADAS PARA EL ONOMASTICÓN DE FUNGRAMKB

Como complemento a la clasificación semántica diseñada en la sección anterior, creemos necesario efectuar una ampliación que complete la taxonomía esbozada. Consideramos que una ampliación de los criterios de clasificación de las reglas constituye un enriquecimiento del procedimiento de análisis a realizar, así como puede facilitarnos la resolución del segundo objetivo de nuestro trabajo: *Analizar los resultados de la implementación de la metodología para confirmar o rebatir la hipótesis de que una estructura formal similar en los constructos elaborados puede indicar una motivación semántica compartida.*

Asimismo, estimamos que la elaboración de una nueva taxonomía que tenga en cuenta criterios más formales que semánticos puede enriquecer el modelo de conocimiento y permitir una implementación de la metodología de creación de reglas

más rica, a la vez que más sistemática. Esta observación proviene del hecho que, mientras se trabajaba en la elaboración de las reglas para las 231 propiedades analizadas, se ha detectado que en numerosas ocasiones existe más de una posibilidad de elaborar la estructura formal de una regla, transmitiendo la misma realidad conceptual. A primera vista, podría parecer una cuestión problemática, ya que implica una duplicidad de estructuras para expresar un mismo concepto, pero bien mirado podría tratarse de todo lo contrario, es decir, de la ventaja que supone para el ingeniero del conocimiento el poder utilizar más de una estructura formal para reflejar un mismo constructo conceptual. Esta situación podemos ilustrarla mediante un ejemplo matemático basado en la propiedad conmutativa, es decir, que la suma de $2+2$ arroja el mismo resultado que la de $3+1$, la de $1+3$ o incluso la de $1+1+1+1$. Ahora bien, puesto que estamos tratando con el particular elemento que es el lenguaje natural, cierto es que algunos autores rechazan la existencia de la sinonimia absoluta, además de que en muchas ocasiones, la forma también constituye una fuente de significado. A este respecto, debemos decir que la elección de una u otra estructura formal puede venir determinada por la intención del ingeniero del conocimiento de reflejar un matiz de contenido de una manera particular. Sin embargo, debemos recordar que estamos tratando con COREL, una interlengua compuesta de conceptos unívocos y no de lenguaje natural, el cual tiene una tendencia manifiesta a la polisemia auspiciada por el principio de economía lingüística. Además, el equilibrio en el grado de granularidad que la descripción en COREL posibilita y el cual opera a nivel global en FunGramKB permite que las diferencias formales o estructurales no constituyan una obstrucción a la homogeneidad de la base, ni al producto resultante de la codificación de las reglas en COREL. Dicho de otro modo, y retomando el ejemplo matemático anterior: si

utilizamos COREL, el resultado de la suma siempre será 4, al margen de los elementos que la compongan.

Así pues, a continuación presentamos una nueva propuesta de clasificación de las reglas del Onomasticón de FunGramKB, atendiendo a criterios más formales que semánticos, si bien nunca se deben perder de vista ambas dimensiones.

5.4.1. LA CLASIFICACIÓN FORMAL DE LAS REGLAS DEL ONOMASTICÓN: TIPOS DE ESTRUCTURAS

Dada la gran variedad de reglas y combinaciones de conceptos posibles, efectuar una taxonomía formal a base de compartimentos cerrados en los que se encasillen reglas completas, como se ha hecho en la clasificación semántica, no parece ser una opción útil o factible en este caso. El motivo es que dicha práctica arrojaría o bien un número demasiado elevado de tipos, dadas las múltiples combinaciones potenciales o, por el contrario, una generalización demasiado amplia y poco concreta de las estructuras. Por tanto, creemos necesario ir más allá de una clasificación taxonómica como la que hemos presentado en la sección anterior, basada en tipos de reglas completas, en favor de una tipología más abierta, basada en los diversos tipos de estructuras formales que pueden mostrar las predicaciones en cada una de las reglas. Especialmente, la nueva taxonomía formal que mostramos está basada en la colocación de los elementos <C> y <VAL> y la manera en que ambos se relacionan en el ámbito de cada regla concreta, tanto entre ellos como con otros elementos de la misma.

A continuación, procedemos a realizar una descripción de las construcciones formales o estructuras presentes en diversas reglas, así como de su utilidad a la hora de servir de apoyo a la metodología de creación de las mismas descrita en el Capítulo IV. Finalmente, al término de esta sección mostramos una tabla resumen que contiene dicha

clasificación formal en su totalidad, junto a las características principales de cada tipo de construcción y un ejemplo de cada uno de ellos, así como una tabla de correspondencias entre los tipos de estructuras formales y su grado de aparición, expresado como un porcentaje, en los cuatro tipos semánticos de propiedades.

5.4.1.1. La estructura CONTEXT

La función de la estructura CONTEXT, la cual se encuentra en la mayoría de las predicaciones (e1), es contextualizar la regla, de tal forma que se refleje el vínculo que existe entre la entidad <C> y la clase o concepto superordinado al que pertenece. Un ejemplo de esta estructura es la predicación (e1) de las reglas pertenecientes a la clase “Place”, que en lenguaje natural se traduciría como *“existe una entidad <C> y esta entidad es siempre un lugar”*:

(13)

+ (e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent)

No obstante, gracias al mayor número de reglas elaboradas en el presente trabajo, podemos observar variantes más elaboradas de esta estructura. Es el caso de las propiedades de la clase “Work”. En esta ocasión, para ilustrar la predicación de contexto, debemos utilizar la estructura algo más compleja que observamos a continuación:

(14)

+ (e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 |
+INFORMATION_OBJECT_00)Referent)

A pesar de que la predicación en (14) mantiene la misma estructura de papeles temáticos del tipo “Theme + Referent”, advertimos que el argumento “Referent” del elemento x2 no es un concepto único, sino que es expresado mediante la inclusión de

dos conceptos dentro del argumento x_2 , los cuales están unidos a través del operador lógico de disyunción inclusiva $|$. Esta estructura surge como consecuencia de la necesidad de reflejar una dimensión conceptual que en la Ontología de FunGramKB está dividida entre dos conceptos. Por un lado, encontramos el concepto +ARTIFICIAL_OBJECT_00, el cual debería ser de manera natural el concepto más cercano a la clase “Work” de DBpedia, puesto que su descripción en lenguaje natural es “*an object that is made by a person*”, según nos indica la Ontología de FunGramKB. Sin embargo, hacer uso únicamente de este concepto en la predicación que estamos elaborando hubiera supuesto dejar fuera de este constructo conceptual a una parte importante de las obras creadas por una persona, como se trata de cualquier tipo de obra escrita (libros, poemas, canciones, etc.), ya que dichos conceptos se encuadran bajo el hiperónimo +INFORMATION_OBJECT_00 de la Ontología de FunGramKB. Si bien sería deseable el hecho de reorganizar en la Ontología de FunGramKB la dimensión conceptual +INFORMATION_OBJECT_00, con objeto de encajar las entidades de esta categoría bajo la jerarquía o bien de los objetos naturales o bien de aquellos creados por las personas (+NATURAL_OBJECT_00 o +ARTIFICIAL_OBJECT_00, respectivamente), en el momento actual no se ha acometido aún dicha reestructuración, por lo que no debemos cerrar el acceso a una importante parte de las entidades de Wikipedia mediante la ausencia de la dimensión de las obras escritas. Por tanto, se hace necesaria la inclusión tanto del concepto +ARTIFICIAL_OBJECT_00 como de +INFORMATION_OBJECT_00 en la predicación de contexto de todas las propiedades atribuidas a la clase “Work”.

5.4.1.2. La estructura ATTRIBUTE

Las estructuras ATTRIBUTE suelen presentar el patrón “Theme + Attribute + Quantity”, siendo “Quantity” reflejado a través de un satélite (participantes de tipo “P”)

en el que se integra el elemento <VAL> como operador. Encontramos ejemplos de esta estructura en la mayoría de las predicaciones (e2) de las reglas de naturaleza semántica atributiva, por ejemplo, en la siguiente predicación de la propiedad *width* de la clase “Place”:

(15)

**(e2:+BE_01(x1)Theme (x3:+WIDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)*

El equivalente en lenguaje natural de esta predicación sería “*la entidad (x1) posee una propiedad que es la anchura y que alcanza el valor de <VAL> metros*”. Por tanto, encontramos que la estructura ATTRIBUTE es muy productiva a través de las diferentes clases de DBpedia contenidas en el Onomasticón, ya que podemos utilizarla en una multitud de reglas de diversas clases, simplemente cambiando el concepto que se incluye dentro del argumento etiquetado como “Attribute”.

5.4.1.3. La estructura EV+SAT

Una vez que hemos retratado las dos estructuras más comunes que aparecen en la gran mayoría de las reglas de la clase “Place”, procedemos a analizar otro tipo de estructuras presentes en diferentes clases. Así pues, comenzamos tomando como ejemplo la propiedad *birthDate* de la clase “Person”:

Propiedad	<i>birthDate</i>
Clase	Person
Tipo de bio-estructura	Story
Regla en COREL	<i>+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) + (e2: ing +LIVE_00(x1)Theme (f1: <VAL> +DATE_00)Time)</i>

Tabla 9. La propiedad *birthDate* de la clase “Person”.

A la luz de esta propiedad, debemos tener en cuenta varias características destacables. Si observamos la predicación (e1), enseguida detectamos que se trata de una estructura del tipo CONTEXT, tanto por su construcción formal como porque su función es la de efectuar el vínculo contextual entre la entidad <C> y la clase a la que pertenece (i.e. una persona, +HUMAN_00). A continuación, observamos en la predicación (e2) que ya no estamos frente a una estructura atributiva, sino que nuevos conceptos aparecen en ella. A nivel semántico, nos encontramos frente a una regla del tipo *predicativa intrínseca*, ya que el contenido semántico de la propiedad (es decir, la fecha de nacimiento de una persona) es una predicación o secuencia de hechos personal e íntima a dicha entidad, la cual puede ser expresada en COREL sin necesidad de la intervención de entidades adicionales y cuya interpretación no depende de la interrelación con ellas. Con respecto a los rasgos formales de la predicación, lo primero que nos llama la atención es el uso de un operador aspectual colocado frente al evento +LIVE_00, el operador *ing*. Este operador aspectual se corresponde con el aspecto ingresivo de una actividad y, como hemos podido observar hasta ahora, su inclusión no constituye una práctica extendida a la totalidad de las reglas del Onomasticón. Sin embargo, en ocasiones resulta necesaria la inclusión de un operador aspectual en las predicaciones, especialmente si su inserción o ausencia constituye una modificación significativa del contenido conceptual de la predicación. En este caso, la inclusión del operador de aspecto ingresivo resulta vital (nunca mejor dicho), ya que completa el contenido conceptual del evento +LIVE_00 para conferirle el significado de “nacer” (i.e. *comenzar* a vivir) en una fecha (+DATE_00) determinada (<VAL>) y no simplemente *estar vivo*. Otro ejemplo a través del cual podemos comprobar la importancia de los operadores aspectuales es la propiedad *subsequentWork* de la clase “Work”:

Propiedad	<i>subsequentWork</i>
Clase	Work
Tipo de bio-estructura	Story
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+CREATE_00 (x3)Theme (x4:<VAL>)Referent (f1:(e3: past +CREATE_00 (x3)Theme (x1)Referent))Condition)

Tabla 10. La propiedad *subsequentWork* de la clase “Work”.

Como podemos ver en la Tabla 10, el éxito de la representación conceptual de la propiedad *subsequentWork* depende casi por completo del uso correcto de los operadores aspectuales, ya que el rasgo principal que se pretende trasladar mediante esta regla es una característica de naturaleza temporal, algo tradicional de las bio-estructuras del tipo *historias*. En este caso, el elemento <VAL> designa a un trabajo realizado con posterioridad a otro trabajo previo, representado por el elemento <C>. Para transmitir estos matices de significado, nos valemos del operador de temporalidad *past*, gracias al cual podemos plasmar el marco temporal de esta propiedad. De este modo, en la predicación (e2) se posibilita la representación conceptual correcta en la que se entiende que el elemento <VAL> es creado una vez que se cumple la condición de que el elemento <C> ha sido creado previamente por esa misma persona.

Asimismo, retornando a la predicación (e2) de la propiedad *birthDate* (Tabla 9), observamos que el elemento <VAL> se encuentra incrustado dentro de un satélite (f1), algo que constituye una práctica común a lo largo de las reglas del Onomasticón de FunGramKB. Si bien no parece productivo considerar la inclusión de los operadores aspectuales arriba mencionados como un rasgo identificativo de un tipo de predicación

per se, a pesar de la utilidad que pueden tener de manera puntual, sí que consideramos la estructura “Event + Participants + Satellite”, en la que el elemento <VAL> aparece dentro del satélite, una estructura tremendamente productiva a través del Onomasticón, por lo que denominamos a este tipo de estructura EV+SAT. Un ejemplo de esta estructura lo encontramos en la segunda predicación de la propiedad *birthDate* ilustrada en la Tabla 9, así como en la segunda predicación de la propiedad *deathPlace* (16) o de la propiedad *knownFor* (17), ambas pertenecientes a la clase “Person”:

(16)
 +(e2: +DIE_00 (x1)Theme (f1: <VAL>)Location)

(17)
 *(e2: +KNOW_00 (x3)Theme (x1)Referent (f1: <VAL>)Reason)

Asimismo, encontramos que la estructura EV+SAT también puede contemplar la inclusión de una predicación incrustada dentro de un argumento o de un satélite, como sucede en el argumento (x4) de la propiedad *day* de la clase “City”:

Propiedad	<i>day</i>
Clase	City
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+CITY_00)Referent) *(e2: +SHOW_00 (x3)Theme (x4: (e3: +RESPECT_00 (x5)Referent (x2) Theme))Referent) (f1: <VAL> +DATE_00)Time)

Tabla 11. La propiedad *day* de la clase “City”.

Por tanto, debido a su flexibilidad y versatilidad, también podemos afirmar que la estructura EV+SAT resulta bastante eficiente para la creación de reglas en el Onomasticón.

5.4.1.4. La estructura EVENT ROLE

A continuación, vemos un nuevo tipo de estructura que, si bien resulta algo similar a la estructura EV+SAT que acabamos de mostrar, debido a su sencillez resulta todavía más simple, puesto que se basa en una construcción que posee un rasgo hasta ahora no demasiado extendido en el Onomasticón: la predicación no contiene ningún satélite, por lo que el elemento <VAL> se integra como un argumento más del evento reflejado. Para ilustrar de manera más clara esta estructura, mostramos la regla para la propiedad *residence*:

Propiedad	<i>Residence</i>
Clase	Person
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +LIVE_01 (x1)Theme (x3:<VAL>)Location)

Tabla 12. La propiedad *residence* de la clase “Person”.

La segunda predicación de la propiedad *residence* es capaz de ilustrar, sin mayor instrumento que un evento y los participantes en el mismo (argumentos), el constructo conceptual reflejado en *residence*, confiriendo al elemento <VAL> su integración como un papel temático más. Por tanto, a este tipo de estructura la podemos denominar EVENT ROLE BARE, ya que no precisa de mayor adición para reflejar la propiedad que deseamos codificar.

Sin embargo, además de la sencilla estructura EVENT ROLE BARE, hallamos otra variante de la estructura EVENT ROLE que también se encuadraría en esta categoría. Se trata de una estructura en la que, además de encontrarnos con el esquema “Event + Participants”, encontramos nuevamente elementos satélites, si bien en ninguno de ellos se inserta el elemento <VAL>. Es importante efectuar una diferenciación clara de esta nueva estructura con respecto a la presentada en la tipología EV+SAT, puesto que la separación entre ambas viene marcada por el contenido del satélite: en EV+SAT el elemento <VAL> siempre aparece dentro del satélite, mientras que en esta nueva estructura, que denominamos EVENT ROLE COMPLEX, para diferenciarla de su categoría hermana EVENT ROLE BARE, los satélites que aparecen son meramente informativos o utilizados para completar la información conceptual de la predicación, pero en los que nunca se incluyen los elementos <C> o <VAL>. Un ejemplo de la estructura EVENT ROLE COMPLEX lo encontramos en la siguiente predicación, perteneciente a la regla *neighboringMunicipality* de la clase “Municipality”:

(18)

*(e2:+BE_02 (x3: <VAL>)Theme (x1)Location (f1:+NEAR_00)Position)

Como observamos en esta predicación, el satélite (f1) se limita a completar la información conceptual del evento, con el fin de concretar el lugar donde se encuentra el municipio del que se esté hablando, sin incluir dentro de ella a los elementos <C> o <VAL>.

5.4.1.5. Las estructuras TANDEM-CROSS REF, EMBED CONDITION y RANGE-IN-SAT: las preferencias de selección

En ocasiones, sucede que podemos codificar de diversas maneras una misma dimensión conceptual, lo que ya hemos sugerido que no debemos interpretar como una falta de homogeneidad o inconsistencia, sino como una ventaja que nos permite

combinar con flexibilidad elementos de diferentes maneras para obtener un resultado óptimo. Este dilema suele aparecer en aquellas reglas en las que debemos hacer uso de algún mecanismo que nos permita precisar la naturaleza concreta de alguno de los participantes en un evento, ya que de otro modo estaríamos dejando incompleta dicha dimensión o, incluso, fomentando una incorrección. Este fenómeno es el equivalente a las preferencias de selección que deben respetarse para algunos eventos en la Ontología de FunGramKB. Esto es, por ejemplo, si en el marco temático de un evento, un participante o argumento viene definido obligatoriamente como +HUMAN_00 o +ANIMAL_00, significa que el papel temático en el evento en cuestión sólo puede ser desempeñado o bien por un humano, o bien por un animal. Un buen ejemplo de ello son los eventos de naturaleza biológica como comer, tragar, respirar, etc. (+EAT_00, +SWALLOW_00, +BREATHE_00, éste último incluso reservado únicamente para los animales mamíferos). De igual forma, y a pesar de que en el Onomasticón no existe la figura de los marcos temáticos, en las reglas almacenadas en él también encontramos predicaciones en las que es preciso definir de manera clara la naturaleza de alguno de los participantes en el evento, puesto que de otro modo estaríamos excluyendo información esencial para el buen funcionamiento de la regla en el Onomasticón. Es decir, debemos consignar esta información adicional para posibilitar con éxito el traslado de la información desde las fuentes externas (p. ej. DBpedia) a FunGramKB. Para ilustrar este fenómeno con mayor claridad, observemos las siguientes reglas elaboradas para las propiedades *locationCity* y *keyPersonPosition* de la clase “Organisation”:

Propiedad	<i>locationCity</i>
Clase	Organisation
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +CITY_00)Attribute)

Tabla 13. La propiedad *locationCity* de la clase “Organisation”.

Propiedad	<i>keyPersonPosition</i>
Clase	Organisation
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+WORK_01 (x3)Theme (f1: x1)Location) *(e3:+BE_01 (x3)Theme (x4:m +IMPORTANT_00)Attribute) *(e4: +DO_00 (x3)Theme (x5:<VAL>)Referent (f2: (e5:+BE_00(x5)Theme (x6:+JOB_00)Referent))Condition)

Tabla 14. La propiedad *keyPersonPosition* de la clase “Organisation”.

En las reglas ilustradas en la Tabla 13 y Tabla 14, observamos que hay ciertas preferencias de selección que deben ser respetadas con objeto de elaborar de manera correcta la regla correspondiente. En *locationCity*, es necesario indicar que la ubicación (“Location”) de la organización es una ciudad y no, por ejemplo, un país o una dirección postal⁸³. La manera en que se ha codificado dicha condición es a través de la inclusión de la predicación (e3), la cual no es rebatible, lo que se revela mediante el símbolo + al inicio de la predicación. De este modo, queda manifestado claramente que

⁸³ Cf. con la propiedad *residence* mostrada en la Tabla 12, donde no es necesaria especificación alguna de preferencias de selección.

el argumento (x3: <VAL>) debe ser una ciudad, +CITY_00. En este tipo de estructura, otra característica que llama la atención con respecto a la estructura global de la regla es que el elemento <VAL> ya no aparece de manera obligatoria en la última predicación de la misma, como suele ser habitual, sino que se mantiene en la predicación intermedia, ya que la última predicación permite la definición de las preferencias de selección del elemento que sea preciso acotar.

Sin embargo, la riqueza del lenguaje COREL nos permite también efectuar la clarificación de preferencias de selección a través de otra construcción: la inclusión de un satélite de condición que especifique alguna característica del elemento <VAL>. De este modo, a través de este tipo de satélite, se materializa la preferencia de selección del participante que así lo precise. Por ejemplo, en la propiedad *keyPersonPosition* ilustrada en la Tabla 14, se ha decidido utilizar esta estructura para reflejar que la actividad que la persona clave en una empresa desarrolla ha de ser un puesto de trabajo dentro de dicha organización. Asimismo, se ha intentado ir un paso más allá en la elaboración formal de esta plantilla, ya que dentro del satélite (f2) está contenida una predicación completa, (e5), que refleja las condiciones que debe cumplir el elemento <VAL> de esta regla.

Así pues, a efectos de otorgar una nomenclatura en nuestra clasificación a estas dos posibilidades estructurales, podemos optar por denominar TANDEM CROSS-REF a la opción consistente en añadir una predicación adicional, siempre vinculada a otra anterior a través de una co-indización de los argumentos de la primera, a modo de referencia cruzada, que efectúa una preferencia de selección de algún elemento de la predicación previa. En cuanto al segundo caso, podemos denominarlo EMBED CONDITION, ya que dicha preferencia de selección se refleja a través de una predicación empotrada dentro de un satélite de condición.

Ahora bien, retomando las posibles opciones que COREL nos ofrece para reflejar las preferencias de selección dentro de una regla, encontramos una tercera opción que nos permitiría indicar estas preferencias sin necesidad de emplear una estructura TANDEM CROSS-REF o una del tipo EMBED CONDITION. Esta tercera opción está conectada a la elección de las unidades de medida que hemos mencionado en la metodología de creación de reglas enumerada en el capítulo anterior, concretamente en el paso 3: *“En caso de valor numérico, identificación de la unidad de medida en la que se expresa el valor de la propiedad”*. Si reflexionamos acerca de las unidades de medida que deben ser consignadas en las reglas cuando se trata de propiedades que ostentan valores numéricos, podríamos afirmar que no se trata más que de otro tipo de preferencia de selección, ya que su función es la de especificar qué tipo de elemento se cuantifica mediante el valor que se proyecte en <VAL> y cuál es su naturaleza conceptual (i.e. si el número que aparece en el lugar de <VAL> son metros, metros cuadrados, segundos, etc.).

Del mismo modo, podemos plantearnos qué sucede con aquellas propiedades que no ostentan propiedades numéricas pero que, presentando una estructura en la que el elemento <VAL> se encuentra incrustado dentro de un satélite, como ocurre en las estructuras del tipo ATTRIBUTE o EV+SAT, precisan de una especificación conceptual mayor sobre la naturaleza de este elemento <VAL>. La alternativa que hemos encontrado para este tipo de casuística es la de incorporar el concepto que actúa como preferencia de selección dentro del propio satélite, en aquellos casos en los que no podemos actuar mediante alguna de las otras dos opciones. Es decir, podemos optar por esta nueva alternativa cuando no podemos utilizar una estructura TANDEM CROSS-REF, ya que este tipo de estructura se basa en la co-indización de los argumentos entre predicaciones para funcionar, y un satélite que contiene el elemento <VAL> en

aislamiento ya no admite un índice dentro de él. Asimismo, lo podemos emplear cuando tampoco podemos utilizar la estructura EMBED CONDITION, puesto que no es posible incluir dentro de un satélite el elemento <VAL> seguido inmediatamente de una predicación. El motivo es que este tipo de estructuras no son válidas en COREL, ya que <VAL> no puede ocupar el lugar de un operador de predicación⁸⁴ e ir al frente de una predicación, un satélite o un argumento. Por tanto, la tercera opción que empleamos en este tipo de casos es la de incluir el concepto dentro del propio satélite. Puesto que la unidad de medida también se denomina “range” en DBpedia en diversas ocasiones y con objeto de diferenciar esta estructura de las simples indicaciones de unidades de medida en las propiedades con valores numéricos, denominamos a este tipo de estructura RANGE-IN-SAT. Un ejemplo de esta estructura lo encontramos en la propiedad *language* de “Populated Place” o en la propiedad *architecturalStyle* de la clase “Building”:

Propiedad	<i>language</i>
Clase	Populated Place
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2: +LIVE_01 (x3:+POPULATION_00)Theme (x1)Location) *(e3:+SAY_00 (x3)Theme (x4)Referent (x5)Goal (f1: <VAL> +LANGUAGE_00)Instrument)

Tabla 15. La propiedad *language* de la clase “Populated Place”.

⁸⁴ La gramática de COREL permite que sólo los operadores de predicación de probabilidad, de modalidad y epistémicos puedan colocarse al frente de una predicación.

Propiedad	<i>architecturalStyle</i>
Clase	Building
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00(x1: <C>))Theme(x2:+BUILDING_00)Referent) +(e2:+BE_01(x1)Theme(x3:<VAL> +STYLE_00)Attribute)

Tabla 16. La propiedad *architecturalStyle* de la clase “Building”.

No obstante, en ocasiones, la necesidad de indicar alguna preferencia de selección en una regla concreta del Onomasticón viene también condicionada por un elemento externo a la regla, el cual tiene que ver con su relación con las demás propiedades de una clase. En otras palabras, si detectamos una especificación mayor entre propiedades de una misma clase, nos veremos obligados a plasmar dicha especificación en las reglas que elaboremos, ya que de otro modo podríamos incurrir en confusión o incluso incorrección. Sin embargo, si el catálogo de propiedades para una clase carece de sobre especificación conceptual, entonces parece posible adoptar un enfoque algo más laxo con respecto a la especificación de preferencias de selección. Por ejemplo, en la clase “Organisation” encontramos tres propiedades diferentes para designar la ubicación de una organización: *location*, *locationCity* y *locationCountry*. Este hecho nos aboca a tener que especificar necesariamente en cada una de las reglas creadas para esa clase si se trata de un lugar genérico (satélite “Location”), una ciudad (+CITY_00) o un país (+COUNTRY_00), lo que en los dos últimos casos hemos decidido reflejar a través de la construcción TANDEM-CROSS REF. La diferencia entre las reglas creadas para estas tres propiedades podemos observarla a continuación:

Propiedad	Clase	Tipo de bio-estructura	Regla en COREL
<i>location</i>	Organisation	Snapshot	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location)
<i>locationCity</i>	Organisation	Snapshot	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +CITY_00)Attribute)
<i>locationCountry</i>	Organisation	Snapshot	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +COUNTRY_00)Attribute)

Tabla 17. Las propiedades *location*, *locationCity* y *locationCountry* de la clase “Organisation”.

Por contra, encontramos propiedades en las que no es necesario sobre-especificar la naturaleza de los elementos que van a ser proyectados en el elemento <VAL>, ya que no se encuentran otras propiedades dentro de la misma clase con las que pueda existir algún conflicto conceptual. Esto sucede en el caso de la propiedad *publisher*, la cual hemos observado en la Ilustración 16 en el Capítulo IV. A pesar de que la etiqueta “range” de esta propiedad indica que se trata de una empresa (“Company”, como indica la información que nos aporta DBpedia y también Schema.org), para reflejar lo cual necesitaríamos añadir una predicación adicional del tipo TANDEM CROSS-REF, EMBED CONDITION o RANGE-IN-SAT, no se ha estimado preciso hacerlo. El motivo es que en este caso, a la luz del resto de propiedades para la clase “Work”, no existe ambigüedad ni problemas de interpretación. Otro caso diferente sería si existiera más de una regla para reflejar al artífice de la publicación de una obra, como por ejemplo *publisher* y otra hipotética propiedad denominada *publisherCompany*. En este supuesto, entendemos que sí sería necesario efectuar una distinción a través de las preferencias de selección, ya

que de otro modo estaríamos propiciando una ausencia de completitud conceptual. Sin embargo, como en este caso solo aparece la propiedad *publisher*, el algoritmo de recuperación de información funcionaría de manera correcta de todos modos.

Igualmente, otras consideraciones al respecto están relacionadas con lo que ya indicábamos en el Capítulo IV sobre esta propiedad, es decir, el hecho de que el concepto \$PUBLISH_00 no se corresponde con la definición “lanzar al mercado”, que es lo que hace una editorial según la información contenida en DBpedia y Schema.org, sino que, en la Ontología de FunGramKB, el concepto \$PUBLISH_00 implica “escribir algo y después lanzarlo al mercado”, siendo el mismo argumento (x1) quien realiza ambas cosas, lo que hemos solucionado mediante la introducción del satélite “Company”, como ya hemos visto en el capítulo anterior.

En otra regla también vinculada al concepto \$PUBLISH_00, denominada *publishDate*, no se ha especificado tampoco ninguna preferencia de selección acerca de la naturaleza del agente de la acción, con el propósito de dejarlo abierto, ya que en este caso no sabemos si es el propio escritor o autor quien publica su trabajo o si lo hace otra persona o una empresa. En el caso de dicha propiedad, creemos más adecuado centrar el eje conceptual de la regla sobre el evento de publicar y en la fecha cuando esto acontece, más que en una sobreespecificación de los participantes concretos que intervienen en el acto.

5.4.1.6. La estructura GENITIVE

Finalmente, otra estructura bastante popular en algunas clases concretas del Onomasticón (i.e. en la clase “Person”) es aquella que se utiliza para expresar las relaciones entre entidades, especialmente las relaciones interpersonales. Si bien a nivel semántico no podríamos hablar de una relación de posesión, a nivel sintáctico resulta menos problemático hacerlo, ya que este tipo de estructuras se fragua dentro de este

marco de pertenencia. En nuestra clasificación semántica, este tipo de estructura aparece típicamente dentro de las reglas denominadas *predicativas relacionales*. Un ejemplo de este tipo de relaciones lo encontramos en diversas propiedades de “Person”: *father*, *mother*, *child*, *partner*, *spouse*, *relative*, etc., donde intervienen dos entidades principales con un vínculo a nivel conceptual entre ellas: por un lado, la entidad <C> y, por otro, la entidad <VAL>, siendo ambas personas y estando relacionadas a través del parentesco. Si bien, en principio, la manifestación de este constructo conceptual parece seguir una estructura del tipo EVENT ROLE COMPLEX, creemos que esta especial conexión entre <C> y <VAL> merece ser detallada en una categoría aparte. El elemento que se utiliza para reflejar esta relación entre entidades es un satélite de referencia (“Referent”) hacia el argumento (x1), que es tradicionalmente el que ostenta el valor <C>, expresado como “(f1:x1)Referent”. Con objeto de reflejar formalmente el vínculo entre la entidad <C> y la otra entidad presente en la regla elaborada, nos valemos de la siguiente estructura, que observamos en la propiedad *father*:

Propiedad	<i>father</i>
Clase	Person
Tipo de bio-estructura	Snapshot
Regla en COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +BE_01 (x3: <VAL>)Theme (x4: +FATHER_00)Attribute(f1:x1)Referent)

Tabla 18. La propiedad *father* de la clase “Person”.

Puesto que esta estructura sirve para vincular la relación de posesividad sintáctica entre dos entidades, de igual modo que lo haría el genitivo, la denominamos GENITIVE.

Finalmente, como recapitulación de la tipología que hemos descrito en esta sección, mostramos una tabla resumen con las diferentes estructuras identificadas y las características que las definen:

Tipos de estructuras atendiendo a criterios de clasificación FORMALES	
CONTEXT	<p>Función: contextualizar la regla y vincular a la entidad <C> con la clase a la que pertenece.</p> <p>Ejemplo: + (e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent)</p>
ATTRIBUTE	<p>Función: Mostrar un atributo de la entidad <C> y cuantificarlo a través del valor <VAL> en la construcción típica “Theme + Attribute + Quantity”, si bien a veces admite otro tipo de satélite (por ej. “Time”). La variable <VAL> debe aparecer, en todo caso, dentro del satélite.</p> <p>Ejemplo: *(e2:+BE_01(x1)Theme (x3:+WIDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)</p>
EV+SAT	<p>Función: Mostrar un evento realizado por <C> o por otro elemento en el marco de la propiedad y aportar información circunstancial del mismo dentro de un satélite, a través del valor <VAL> (p. ej. lugar, tiempo, modo, causa, etc.).</p> <p>Ejemplo: *(e2: +KNOW_00 (x3)Theme (x1)Referent (f1: <VAL>)Reason)</p>
EVENT ROLE	<p>Función: Mostrar un evento realizado por la entidad <C> en el cual interviene el valor <VAL> como un argumento participante en la acción.</p> <p>Pueden ser BARE si sólo contienen el evento y sus argumentos o COMPLEX, en cuyo caso incluyen también satélites en los que no interviene <VAL>, a diferencia de lo que ocurre en la construcción EV+SAT.</p> <p>Ejemplos: BARE: *(e2: +LIVE_01 (x1)Theme (x3:<VAL>)Location) COMPLEX: *(e2:+BE_02 (x3: <VAL>)Theme (x1)Location (f1:+NEAR_00)Position)</p>

TANDEM-CROSS REF	<p>Función: Codificar una preferencia de selección para el elemento <VAL> añadiendo una predicación adicional paralela pero ligada a la anterior, que efectúa una preferencia de selección de algún elemento de la predicación previa.</p> <p>En este caso, el elemento <VAL> ya no aparece necesariamente en la última predicación de la regla.</p> <p>Ejemplo:</p> <p><i>*(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +CITY_00)Attribute)</i></p>
EMBED CONDITION	<p>Función: Codificar una preferencia de selección para el elemento <VAL> a través de una predicación empotrada dentro de un satélite de condición.</p> <p>Ejemplo:</p> <p><i>*(e4: +DO_00 (x3)Theme (x5:<VAL>)Referent (f2: (e5:+BE_00(x5)Theme (x6:+JOB_00)Referent))Condition)</i></p>
RANGE-IN-SAT	<p>Función: Codificar una preferencia de selección para el elemento <VAL> cuando éste aparece dentro de un satélite en las estructuras ATTRIBUTE o EV+SAT.</p> <p>Ejemplo:</p> <p><i>+(e2:+BE_01(x1)Theme (x3: <VAL> +STYLE_00)Attribute)</i></p>
GENITIVE	<p>Función: Vincular mediante una relación sintáctica de posesión a la entidad <C> y a la entidad <VAL>, especialmente cuando ambas entidades se refieren a personas. En realidad, puede considerarse un tipo especial de EVENT ROLE COMPLEX.</p> <p>Ejemplo:</p> <p><i>+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +BE_01 (x3: <VAL>)Theme (x4: +FATHER_00)Attribute(f1:x1)Referent)</i></p>

Tabla 19. Tipos de estructuras atendiendo a criterios de clasificación formales.

Podemos concluir que, efectivamente, la hipótesis enunciada previamente en Carrión Varela (2010) e incluida en el segundo objetivo del actual trabajo quedaría confirmada a través de dichas tipologías, puesto que observando y analizando con detalle cada una de las reglas y las predicaciones que las componen, descubrimos estructuras análogas en todas ellas, lo que ha permitido agruparlas en categorías definidas conforme a criterios tanto semánticos como formales, atendiendo a las particularidades de cada uno de los tipos observados.

De la misma forma, se ha podido comprobar que una similitud formal conlleva una similitud semántica, ya que sucede que ciertos tipos de estructuras formales se dan cita de manera preferente en ciertos tipos semánticos de reglas. Esto es algo que hemos podido observar, por ejemplo, en las reglas que contienen predicaciones del tipo ATTRIBUTE, las cuales suelen aparecer en su mayoría en las reglas de tipo atributivas, ya que aparecen en un 57,89 % de las propiedades atributivas simples y en un 100% de las propiedades atributivas absolutas. También observamos que las predicaciones del tipo GENITIVE se dan cita mayoritariamente en las reglas de tipo predicativas relacionales en las que aparece una relación de parentesco, ya que su porcentaje de aparición en dicha categoría es mayor que en el resto de categorías. Igualmente, hemos observado que la estructura EVENT ROLE se identifica comúnmente con las reglas de tipo predicativas⁸⁵, además de que esta estructura, junto a EV+SAT, resulta ser plurivalente a través de la totalidad de las reglas, debido a que ambas codifican, respectivamente, o bien un elemento satélite que muestra un valor determinado a través de <VAL>, o bien cuando la variable <VAL> actúa como un argumento más en la predicación. Con objeto de ilustrar más claramente todas estas coincidencias, así como para mostrar el resto de porcentajes de aparición, concluimos esta sección aportando una tabla resumen (Tabla 20) en la que podemos observar el porcentaje de aparición de cada tipo de estructura formal en cada uno de los tipos semánticos. También incluimos en dicha tabla un valor, identificado como F (frecuencia), que indica el porcentaje de aparición de cada tipo de estructura formal, calculado sobre el total de propiedades para las cuales hemos elaborado reglas en lenguaje COREL, cuyo número asciende a 128. Por ejemplo, dicho valor F nos confirma que la estructura CONTEXT está presente en el 100% de las propiedades para las que se ha elaborado una regla, mientras que la estructura

⁸⁵ Especialmente, la variante EVENT ROLE COMPLEX tiende a aparecer en las reglas de tipo predicativas intrínsecas.

ATTRIBUTE aparece en el 17,97% de la totalidad de las reglas y así sucesivamente. Esta información puede ser ampliada en el Apéndice II, el cual contiene las tablas de frecuencia y aparición de cada tipología, junto a las estadísticas y datos totales en cifras de la población del Onomasticón realizada para este trabajo.

		TIPOS SEMÁNTICOS				
		ATRIB SIMPLES	ATRIB ABSOL	PRED REL	PRED INT	F^(*)
ESTRUCTURAS FORMALES	CONTEXT	100,00%	100,00%	100,00%	100,00%	100,00%
	ATTRIBUTE	57,89%	100,00%	9,84%	6,67%	17,97%
	EV+SAT	10,53%	0,00%	24,59%	55,56%	32,81%
	EVENT ROLE	26,32%	100,00%	63,93%	44,44%	52,34%
	TANDEM- CROSS REF	0,00%	0,00%	19,67%	15,56%	14,84%
	EMBED CONDITION	0,00%	0,00%	3,28%	2,22%	2,34%
	RANGE-IN-SAT	0,00%	0,00%	6,56%	11,11%	7,03%
	GENITIVE	5,26%	0,00%	8,20%	2,22%	5,47%

^(*)F: frecuencia de aparición calculada sobre el total de las 128 reglas elaboradas en lenguaje COREL

Tabla 20. Frecuencia de aparición de cada estructura formal en los diferentes tipos semánticos de propiedades.

Así pues, una vez analizados en el presente capítulo los resultados de la aplicación de la metodología creada y habiendo logrado el segundo de los objetivos planteados al inicio del actual trabajo, en el próximo capítulo trasladamos la discusión al siguiente paso lógico, el cual es la consecución de nuestro tercer objetivo: *Determinar la aplicación que pueden tener los resultados obtenidos, gracias a la metodología implementada, para resolver problemas lingüísticos que acucian al PLN.*

CAPÍTULO VI

APLICACIONES DEL ONOMASTICÓN PARA RESOLVER PROBLEMAS LINGÜÍSTICOS

6.1. LA AMBIGÜEDAD EN EL LENGUAJE

Un hecho importante tras el cual subyacen muchos de los problemas y cuestiones por resolver por parte de los expertos en PLN es que, para las computadoras, resulta una ardua tarea el poder entender el lenguaje natural. Esta característica obvia, derivada de la ausencia de sentido común por parte de las máquinas, se encuentra principalmente motivada o agravada por una característica innata del lenguaje natural: la ambigüedad. Esta particularidad podría quizás ser consecuencia de la evolución natural que favorece el principio de economía en la lengua, mediante el cual el ser humano es capaz de transmitir el máximo de información posible utilizando el mínimo necesario de signos lingüísticos⁸⁶. Esta ley, por tanto, podría impulsar la existencia de ambigüedad lingüística de varios tipos: ambigüedad semántica de una misma lexicalización, ambigüedad sintáctica o estructural provocada por elipsis y fenómenos análogos o incluso ambigüedades más allá del mero texto y que aparecen a nivel

⁸⁶ Para una discusión sobre fenómenos derivados del principio de economía lingüística (p. ej. la elipsis) remitimos a Hernández Terrés (1984), así como a Paredes Duarte (2008) para un análisis desde una perspectiva diacrónica.

pragmático como, por ejemplo, utilizar una pregunta que se debe interpretar como una petición o incluso una orden (cf. Vázquez, 2009). No obstante, la existencia de la ambigüedad en los términos y los impedimentos que aquella supone frente a ciertos avances del pensamiento lógico es algo que siempre ha estado presente, incluso puesto de manifiesto por filósofos como Hume, ya en el S. XVIII: *“The chief obstacle, therefore, to our improvement in the moral or metaphysical sciences is the obscurity of the ideas, and ambiguity of the terms.”* (Hume, 1910:336).

La resolución de la ambigüedad y, en especial, la ambigüedad estructural constituye uno de los sempiternos campos de trabajo dentro del PLN (Galicía-Haro *et al.*, 2001; Vázquez, 2009). Nuestro trabajo sugiere cómo el conocimiento cultural insertado en FunGramKB puede ayudar a resolver determinados casos de ambigüedad en el lenguaje que pueden resultar problemáticos para el PLN, lo que a lo largo del presente capítulo se encamina hacia cubrir el tercer objetivo enunciado al inicio de esta tesis doctoral, es decir, *Determinar la aplicación que pueden tener los resultados obtenidos, gracias a la metodología implementada, para resolver problemas lingüísticos que acucian al PLN.*

Por ende, a continuación mostramos cómo la inclusión de conocimiento cultural en una base de conocimiento de semántica profunda como FunGramKB puede ayudar a resolver ciertos problemas de comprensión en el PLN, centrándonos en algunos de los fenómenos lingüísticos más comunes que guardan relación con la ambigüedad: la referencia y la correferencia, la anáfora, la metáfora, la metonimia y el particular fenómeno de la *metaftonimia* (Goossens, 1990).

6.2. LA REFERENCIA Y CORREFERENCIA

Es algo indudable que el principal uso de los nombres propios consiste en la designación de entidades en el mundo que conocemos, lo que podemos denominar

referencia. La definición de *referencia* enunciada por Gottlob Frege pone de manifiesto de una manera muy clara esta idea: “*The referent of a proper name is the object itself which we designate by its means; [...].*” (Frege, 1948:213). Por tanto, cuando aludimos a la *referencia* de una entidad, estamos mencionando la capacidad que una lexicalización concreta tiene para evocar en nuestro entendimiento la entidad a la que representa. De este modo, si una entidad puede ser denotada por más de una lexicalización, estaríamos hablando de que ambas lexicalizaciones son expresiones correferentes, fenómeno al que en este capítulo aludimos como *correferencia*.

En el lenguaje natural, el ser humano es capaz de detectar las referencias y correferencias en la comunicación a través de diversos sistemas, ya sean léxicos (i.e. mediante el uso de unidades léxicas concretas, p. ej.: “¿*El rojo es tu coche? Sí, es mi automóvil*”), sintácticos (p. ej. mediante la concordancia verbal), onomásticos (p. ej. utilizando nombres propios) o incluso deícticos (p. ej. con expresiones como “*El jarrón este siempre está estorbando*”⁸⁷), si bien existe un mecanismo principal que opera de manera inconsciente en el hablante: el sentido común y el conocimiento cultural del mundo. De este modo, un hablante cuyo conocimiento cultural sea de un nivel medio, podrá diferenciar claramente a qué se refiere otro hablante si menciona, por ejemplo, al *astro rey* en una conversación acerca del Sistema Solar o a la *Mona Lisa* si se hace alusión a ella en una charla de arte cuyo tema sea *La Gioconda*. La problemática surge cuando se trata de hacer que la máquina sea capaz de detectar a qué entidad del mundo real se refiere una expresión lingüística o, yendo más allá, capacitar a la máquina para que comprenda de manera más completa tanto las inferencias conceptuales como el comportamiento sintáctico de dicha entidad. Comenzamos por ilustrar un ejemplo de referencia en el mundo real:

⁸⁷ Ejemplo obtenido del *Diccionario panhispánico de dudas* de la REAL ACADEMIA ESPAÑOLA (2005), versión electrónica, fecha de consulta 10/05/2014.

(19)

Nintendo's Super Mario has achieved worldwide sales of more than 100 million games, and will soon be immortalised by Bob Hoskins in a Hollywood movie.

(Davies, 2004-).

Si prestamos atención a la entidad aludida al comienzo de la expresión, *Nintendo's Super Mario*, es muy probable que cualquier hablante medio procedente de la cultura occidental sea capaz de identificar a qué se refiere dicha entidad nombrada. Para la máquina, en este caso para la base de conocimiento FunGramKB, el hecho de comprender a qué se refiere dicha expresión completaría su capacidad de razonamiento a la vez que proporcionaría una herramienta de PLN más potente y robusta.

El conocimiento de las entidades en el mundo real se introduce en este ejemplo mediante la vinculación de las entidades nombradas con su naturaleza conceptual, a través de la cohesión que permite el uso del mismo lenguaje (COREL) en todos los módulos conceptuales de FunGramKB. De este modo, la expresión *Nintendo's Super Mario* en el ejemplo (19) se vincula a entidades de la Ontología de FunGramKB del modo que se explica a continuación.

En primer lugar, la información proyectada desde DBpedia en el Onomasticón de FunGramKB se encuadra dentro de la lista de propiedades asignadas a la clase “Organisation”, ya que la empresa *Nintendo* pertenece a dicha clase. A modo de ilustración, podemos observar a continuación la proyección efectuada para los productos que dicha empresa comercializa, junto a su equivalente en lenguaje natural:

(20)

```
+(e1:+BE_00 (x1: %NINTENDO_00)Theme (x2:+ORGANIZATION_00)Referent)
*(e2:+SELL_00 (x1)Agent (x3: %COLOR_TV_GAME_00 &
%GAME_BOY_LINE_00 & %NINTENDO_ENTERTAINMENT_SYSTEM_00 &
%SUPER_NINTENDO_ENTERTAINMENT_SYSTEM_00 & %VIRTUAL_BOY_00
& %NINTENDO_64_00 & %NINTENDO_GAMECUBE_00 &
```

%NINTENDO_DS_FAMILY_00 & %WII_00 & %NINTENDO_3DS_FAMILY_00 & %WII_U_00)Theme (x4)Origin (x5)Goal)

Nintendo es una organización que comercializa los productos Color TV Game, Game Boy Line, Nintendo Entertainment System, Super Nintendo Entertainment System, Virtual Boy, Nintendo 64, Nintendo GameCube, Nintendo DS Family, Wii, Nintendo 3DS Family y Wii U.

De este modo, a través de la predicación (e1) se materializa la vinculación conceptual entre una clase de entidades del Onomasticón de FunGramKB y un concepto de la Ontología de dicha base de conocimiento, +ORGANIZATION_00, lo que aporta solidez y cohesión al modelo y permite la proyección de la información de manera correcta. A través de la predicación (e2) obtenemos la información relativa a los productos comercializados por dicha organización.

Por tanto, no sólo la entidad *Nintendo* mencionada en (19) encuentra su acomodo en el repositorio conceptual de FunGramKB. La entidad nombrada *Super Mario* también se ubica dentro de aquélla, a través de un mecanismo similar al que arriba se ha mostrado para *Nintendo*. En este caso, la entidad *Super Mario* es una creación u obra humana, por lo que pertenece a la clase “Work”. En la siguiente proyección observamos cómo se efectúa el vínculo conceptual entre la compañía *Nintendo* y la obra digital *Super Mario*:

(21)

+ (e1: +BE_00 (x1: %SUPER_MARIO_(SERIES)_00)Theme (x2: +ARTIFICIAL_OBJECT_00 | +INFORMATION_OBJECT_00)Referent)
*(e2: \$PUBLISH_00 (x3)Theme (x1)Referent (f1: %NINTENDO_00)Company)

Super Mario (Series) es una obra lanzada por la entidad Nintendo.

De este modo, la información proyectada en el Onomasticón de FunGramKB desde DBpedia permite la conexión entre la entidad *Nintendo* y la obra digital *Super Mario*, posibilitando así que el discurso mostrado en (19) sea desambiguado por la

máquina, la cual es capaz de procesar que *Super Mario* es un producto creativo de la compañía *Nintendo* y que, además, es comercializado por dicha entidad. De este modo, se permite todo el conjunto de inferencias conceptuales que implica tanto la pertenencia de *Nintendo* a la clase +ORGANIZATION_00 como la de *Super Mario* a la clase “Work”, la cual se desdobra en +ARTIFICIAL_OBJECT_00 e +INFORMATION_OBJECT_00.

Para terminar esta sección, es interesante aludir a otro tipo de referencia cuyo vínculo con el referente concreto que designa en el mundo real resulta algo más complejo para la máquina. Este fenómeno ocurre cuando es necesario efectuar la desambiguación de elementos que no son entidades nombradas como personas, edificios, lugares, empresas, obras creadas, etc., sino elementos de conocimiento cultural general más difícil de concretar en una entidad única, como por ejemplo acontecimientos históricos o fechas ilustres. Sirva como ejemplo el hecho histórico del levantamiento del 2 de mayo de 1808 en Madrid o la caída de las torres gemelas del World Trade Center en Nueva York en septiembre de 2011, eventos que podrían ser necesarios para la comprensión de determinados textos históricos y/o periodísticos y que están dentro del conocimiento almacenado en el Onomasticón. De este modo, la realización lingüística destacada en el texto del ejemplo (22) mostrado a continuación no sería dificultosa de resolver por FunGramKB (i.e. qué sucesos son los del 2 de mayo), gracias al conocimiento enciclopédico proyectado en el Onomasticón acerca de este evento histórico, proyecciones que mostramos en (23) y que acompañamos de sus equivalentes en lenguaje natural:

(22)

La Junta Municipal del distrito de Centro ha organizado el I Certamen de Teatro, dotado con un premio de 500.000 pesetas. El tema de las obras presentadas girará en torno a la historia de la villa de Madrid, y preferentemente sobre la invasión francesa y los sucesos del 2 de mayo.

(REAL ACADEMIA ESPAÑOLA)

(23)

a)

+ (e1:+BE_00 (x1: % LEVANTAMIENTO_DEL_2_DE_MAYO_00)Theme
(x2:\$RIOT_00)Referent)

+ (e2:+DO_00 (x3)Theme (x1)Referent (f1: 02 +DAY_00 & \$MAY_00 & 1808
+YEAR_00)Time)

El Levantamiento del Dos de Mayo es una revuelta que sucedió el 2 de mayo de 1808.

b)

+ (e1:+BE_00 (x1: % LEVANTAMIENTO_DEL_2_DE_MAYO_00)Theme
(x2:\$RIOT_00)Referent)

+ (e2:+DO_00 (x3)Theme (x1)Referent (f1: %MADRID_00)Location)

El Levantamiento del Dos de Mayo es una revuelta que tuvo lugar en Madrid.

c)

+ (e1:+BE_00 (x1: % LEVANTAMIENTO_DEL_2_DE_MAYO_00)Theme
(x2:\$RIOT_00)Referent)

+ (e2:+DO_00 (x3: %ESPAÑA_00)Theme (x1)Referent)

+ (e3:+FIGHT_00(x3)Theme(x4:%PRIMER_IMPERIO_FRANCES_00)Referent)

El Levantamiento del Dos de Mayo es una revuelta en la que el pueblo español se levanta en armas frente al Primer Imperio Francés.

6.3. LA ANÁFORA

Uno de los fenómenos lingüísticos en los que más se ha trabajado en PLN y cuya resolución ha sido buscada por numerosos proyectos es la anáfora⁸⁸. La anáfora es una forma de presuposición que consiste en apuntar a un referente o elemento previo mencionado (Halliday y Hasan, 1976), siendo un fenómeno lingüístico que contribuye en gran manera a la cohesión de un texto, especialmente cuando se trata de la llamada anáfora endofórica (i.e. cuyo antecedente se encuentra dentro del mismo texto). Esto es así ya que permite la continuación lógica y coherente de una secuencia textual, en lengua escrita u oral, sin necesidad de reiterar continuamente las mismas expresiones lingüísticas o lexicalizaciones de un mismo concepto o referente. Según resumen Fan *et al.* (2005), existen tres elementos clave en la construcción de una anáfora, i.e. el antecedente o anclaje, la expresión que se refiere al antecedente y el enlace: *“The object that is being referred to is called the anchor or the antecedent, the expression that refers to the antecedent is called the referring expression, and the association between the referring expression and the anchor is called the link.”* (Fan *et al.*, 2005:153).

Para la ilustración de este fenómeno, así como para mostrar también el tratamiento y la resolución que FunGramKB efectúa de otros fenómenos lingüísticos relacionados, en esta sección utilizamos el término *anáfora* en su concepción más amplia. Esto incluye a otros fenómenos relacionados como, por ejemplo, la catáfora, puesto que el objetivo principal de los ejemplos y descripciones de esta sección es vincular las cadenas de correferencia existentes en el lenguaje natural, tanto las que hacen alusión a elementos previamente mencionados, como sucede en la anáfora, así

⁸⁸ Véase Mitkov (2002) para una exhaustiva revisión de los diversos proyectos encaminados a la resolución de la anáfora a través de la historia.

como aquellas cadenas correferenciales materializadas mediante la mención posterior de la entidad nombrada, lo que conocemos como catáfora.

Si bien a la mayoría de los hablantes nativos de una lengua la identificación y resolución del antecedente al que hace alusión una expresión anafórica les resulta una tarea relativamente sencilla, no siempre sucede de este modo para las máquinas que procesan el lenguaje natural. Con objeto de ilustrar este fenómeno, se pueden observar los siguientes ejemplos en los cuales se han destacado las alusiones a entidades nombradas (antecedentes) y sus correspondientes anáforas:

(24)

*Good morning from Hollywood. You know, here at the famous Grauman's Chinese Theater, they have handprints and footprints from all sorts of famous people, from Jimmy Stewart to the cast of Harry Potter. And this will be the latest, **Michael Jackson**, the imprint of **his** famous sequined glove over there, **his** footprints. And then the handprints of **his** three children who, just hours ago, took steps to ensure **their father's** Hollywood immortality.*

(Davies, 2008-).

(25)

*The Nation has since denied any "official business or professional relationship" with **Jackson**, and yet several of their members, sober-faced and in business suits, lined up outside the Santa Maria courthouse at **the singer's** arraignment last January. More than a thousand **Jackson** supporters thronged the streets in front of the courthouse, holding up signs and cheering as they strained for a glimpse of **Michael, who** was accompanied that morning by **his** mother, father, brother Jermaine and sister Janet. Suddenly a roar went up. There was the **King of Pop** on the roof of a vehicle waving **his** arms and blowing kisses at the screaming crowd.*

(Davies, 2008-).

En (24), se observa que la referencia al fallecido cantante *Michael Jackson* (entidad nombrada) se replica mediante la utilización en tres ocasiones del determinante posesivo *his*, lo que facilita la cohesión textual y permite identificar, aun en el supuesto

caso de ignorar quién es *Michael Jackson*, que *his* se refiere en las tres ocasiones al mismo referente o entidad masculina. Además, la mención posterior como *their father* se debe unir al referente *his three children*, que a nivel conceptual o semántico queda vinculado a *their father*, lo que hace posible interpretar el texto de manera coherente por parte del receptor del mensaje. En caso de ser interpretado por una máquina, concretamente por la base de conocimiento FunGramKB, la interpretación de las expresiones *his* en relación a *their father* contenidas en (24) podría ser resuelta sin mayor problemática, gracias a la carga conceptual y morfológica que poseen los diferentes módulos de la base de conocimiento y que permite, por medio de su razonador, vincular los conceptos entre sí. Por ejemplo, gracias al conocimiento conceptual almacenado en la Ontología, el razonador de FunGramKB sabe que un *padre* es una persona de género masculino que tiene descendencia, como observamos en el postulado de significado del concepto +FATHER_00:

CONCEPTO	+FATHER_00
SUPERORDINADO(S)	+PARENT_00
POSTULADO DE SIGNIFICADO	+(e1: +BE_00 (x1: +FATHER_00)Theme (x2: +PARENT_00)Referent) +(e2: +BE_01 (x1)Theme (x3: +MAN_00)Attribute)
DESCRIPCIÓN	a male parent (also used as a term of address to your father); "his father was born in Atlanta"

Tabla 21. El concepto +FATHER_00 en la Ontología de FunGramKB.

Sin embargo, para identificar el sexo de la entidad *Michael Jackson* y, de este modo, el correspondiente género gramatical que permitiría la vinculación anafórica entre la entidad nombrada, el determinante posesivo *his* y el posterior *their father*, resulta de gran importancia la inclusión en FunGramKB del módulo Onomástico. En este ejemplo concreto, gracias a la importación a través de DBpedia de la información contenida en Wikipedia, es posible saber que la entidad *Michael Jackson* es una persona

padre de tres hijos, como se ilustra a continuación en la proyección de la propiedad *children* para la entidad *Michael Jackson*:

(26)

$(e1: +BE_00 (x1: \%MICHAEL_JACKSON_00)Theme (x2: +HUMAN_00)Referent)$
 $(e2: +BE_01 (x3: \%PRINCE_MICHAEL_JACKSON_I_00 \& \%PARIS-$
 $MICHAEL_KATHERINE_JACKSON_00 \&$
 $\%PRINCE_MICHAEL_JACKSON_II_00)Theme (x4: +CHILD_00)Attribute$
 $(f1:x1)Referent)$

Michael Jackson es una persona cuyos hijos son Prince Michael Jackson I, Paris-Michael Katherine Jackson y Prince Michael Jackson II.

En relación al ejemplo (25), éste pone de manifiesto de manera más explícita la necesidad e importancia de la existencia del Onomasticón. De otro modo, la alusión a *King of Pop* que aparece en dicho texto no podría ser resuelta de manera exitosa por parte de la máquina, la cual, basándose tan sólo en coincidencias léxico-gramaticales, tendría dificultades en determinar qué entidad nombrada masculina mencionada anteriormente en el texto (*father, brother Jermaine, Michael*) desempeña el papel de antecedente de la expresión *King of Pop*. Gracias a la información cultural incluida en el Onomasticón, sabemos que *King of Pop* se refiere a *Michael Jackson*, ya que dicha información aparecerá en la casilla “a/n” (*alternative name*) para la entidad $\%MICHAEL_JACKSON_00$.

Otro ejemplo que ilustra la capacidad de FunGramKB para desambiguar cadenas correferenciales lo podemos obtener de la obra de Mitkov (2002), donde se menciona una cadena de entidades que tienen el mismo referente:

(27)

Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.

(Mitkov, 2002:5)

En (27) podemos observar cómo se efectúa una cadena de correferencias entre *Sophia Loren*, *she* y *the actress*, tal como destaca el autor. Por otro lado, encontramos que entre *Bono* y *U2 singer* también existe una correferencia compartida. Sin embargo, en el caso de que una máquina tuviera que decidir cuáles son las cadenas de correferencia en este extracto, podría existir ambigüedad a la hora de decidir si *the actress* hace referencia a *Sofia Loren* o a *Bono*. No obstante, puesto que dicha información se encuentra dentro de los datos de DBpedia importados al Onomasticón de FunGramKB, de nuevo queda salvaguardada la correcta interpretación referencial, gracias a la proyección de datos relativa a la propiedad *occupation* para la entidad *Sofia Loren* que mostramos a continuación y con la que concluimos esta sección:

(28)

```
+(e1:+BE_00 (x1: %SOFIA_LOREN_00)Theme (x2:+HUMAN_00)Referent)
*(e2: +DO_00 (x1)Theme (x3:+ACTOR_00)Referent (f1:(e3:+BE_00 (x3)Theme
(x4:+JOB_00)Referent))Condition)
```

Sofia Loren es una persona cuya profesión es la de actriz.

6.3.1. LA RESOLUCIÓN DE LA ANÁFORA INDIRECTA

En ocasiones, la coherencia de un extracto textual va más allá de una mera cadena correferencial existente en la misma oración o extracto para desplazarse a referentes más alejados, incluso más allá del emisor, el receptor o los elementos contextuales del momento de la emisión del mensaje. Este fenómeno es lo que se denomina anáfora indirecta. Según Mitkov (2002:15), “*Indirect anaphora arises when a reference becomes part of the hearer’s or reader’s knowledge indirectly rather than by direct mention*”, lo que significa que la conexión de una expresión anafórica con su antecedente resulta más complicada, al no estar basada en una identificación sintácticamente detectable. En cierto modo, podemos establecer una similitud entre esta

noción y la de *deixis en fantasma* que encontramos en Bühler (1985), ya que se refiere a alusiones en el discurso cuyo antecedente debe buscarse en el campo de los recuerdos o de la fantasía, frente otro tipo de referencias como las que puede evocar la *deixis ad oculos* (i.e. antecedentes que emisor y receptor comparten visualmente) o la anáfora común (i.e. aquella evocada por los elementos léxicos del discurso).

Otros autores se refieren a fenómenos relacionados mediante diferentes denominaciones, como por ejemplo Eckert y Strube (1999), que lo denominan *anáfora abstracta*, si bien es cierto que Palomar *et al.* (2000) apuntan que aquéllos aplican este término de una manera más general a cualquier antecedente que no sea un sintagma nominal: “[...], *if the antecedent is a noun phrase then the anaphora is classified as individual anaphora, otherwise, the anaphora is classified as abstract anaphora.*” (Palomar *et al.*, 2000:206). Un nuevo término que podemos relacionar también es el concepto de *bridging* (Martin, 1992), referido a las relaciones que existen entre una anáfora y su antecedente, excluyendo la de identidad, lo que nos permite encuadrar aquí relaciones como, por ejemplo, la hiponimia o la hiperonimia:

“Clark (1977) called bridging descriptions to definite descriptions that either have an antecedent denoting the same discourse entity, but using a different head noun (synonym, hypernym or hyponym) or are related by other relation than identity.”

(Palomar *et al.*, 2000:207).

Debido a la gran complejidad que entraña la resolución de la anáfora indirecta, muchos autores concluyen que: *“Therefore anaphors requiring real-world knowledge for their resolution stand the least chance of being resolved successfully.”* (Mitkov, 2002:34), lo cual parece confirmar que, si en muchos casos este tipo de anáforas pueden resultar complicadas de resolver incluso para un humano que no posea ciertos conocimientos culturales, la resolución por parte de la máquina puede tornarse una quimera. Precisamente por esto, es en estos casos donde el Onomasticón de

FunGramKB puede resultar de especial ayuda para conferir sentido a las referencias expresadas, ya que la integración de conocimiento enciclopédico en este módulo supone un enriquecimiento cultural del modelo de conocimiento, lo que permite la resolución de casos de anáfora indirecta. Observemos el siguiente ejemplo:

(29) When *Take That* broke up, the critics gave *Robbie Williams* no chance of success.
(Mitkov, 2002:15)

En el ejemplo (29), la información acerca de lo que es la entidad *Take That* es necesaria para comprender dónde está la correlación o coherencia semántica de aludir a otra entidad no mencionada previamente, *Robbie Williams*, ya que a simple vista no parece establecerse ninguna cadena correferencial. Sin embargo, la motivación de mencionar a la entidad *Robbie Williams* en este ejemplo responde a su condición de integrante de la banda musical inglesa *Take That*, una información que el Onomasticón de FunGramKB sí es capaz de aportar acerca de la banda y sus integrantes, gracias a la proyección de datos que tiene lugar en dicho módulo, como observamos a continuación:

(30)
+(e1:+BE_00 (x1: %TAKE_THAT_00)Theme (x2:+PEOPLE_00)Referent)
+(e2: +DO_00 (x1)Theme (x3:+MUSICIAN_00)Referent)
*(e3:+COMPRISE_00 (x1)Theme (x4: %GARY_BARLOW_00 &
%HOWARD_DONALD_00 & %JASON_ORANGE_00 & %MARK_OWEN_00 &
%ROBBIE_WILLIAMS_00)Referent)

Take That es un grupo de personas que son músicos, compuesto por Gary Barlow, Howard Donald, Jason Orange, Mark Owen y Robbie Williams.

Otra instancia más compleja de cómo FunGramKB es capaz de resolver este problema se puede observar en el ejemplo que mostramos en (31). Para poder proporcionar una respuesta exitosa a la desambiguación anafórica del discurso, es

necesario que la base de conocimiento posea información relativa a eventos de corte histórico, lo que va más allá de un conocimiento básico acerca de entidades nombradas individuales para referirse a eventos complejos de la historia, conocidos por los hablantes medios de una comunidad pero ignorados, en su gran mayoría, por las máquinas. Observemos el pronombre subrayado en el siguiente ejemplo:

(31)

It is possible that some human should have climbed the World Trade Center Towers without ropes before they were destroyed.

(King, 2013).

En este caso, para que la máquina sea capaz de efectuar la resolución anafórica del antecedente del pronombre *they*, es necesario que aquélla posea conocimiento acerca de que las torres del World Trade Center fueron destruidas en una fecha precisa, ya que tanto los humanos a los que se menciona en el mismo ejemplo como las propias cuerdas utilizadas para escalar podrían ser tanto sintáctica como conceptualmente potenciales antecedentes, puesto que todas ellas son entidades que también pueden ser destruidas, según la información conceptual contenida en la Ontología de FunGramKB. Asimismo, la máquina podría entender que el antecedente más plausible serían las cuerdas, debido a que se trata de la referencia más cercana en el discurso al pronombre *they*. Sin embargo, el Onomasticón proporciona el vínculo conceptual con el conocimiento enciclopédico necesario para desambiguar el antecedente del pronombre *they*, como se ilustra en la proyección de la propiedad *destructionDate* que exponemos a continuación:

(32)

`+(e1:+BE_00 (x1: %WORLD_TRADE_CENTER_TOWERS_00)Theme
(x2:+BUILDING_00)Referent)`

`+(e2: +DESTROY_00 (x3)Theme (x1)Referent (f1: 11+DAY_00 & $SEPTEMBER_00
& 2001 +YEAR_00)Time)`

Las Torres del World Trade Center son unos edificios que fueron destruidos el 11 de septiembre de 2001.

Por tanto, este vínculo conceptual permite que el ejemplo (31) se resuelva de una manera más clara, de forma que se identifique que el antecedente del pronombre *they* es la entidad *World Trade Center Towers*, lo que pone de manifiesto la capacidad de la base de conocimiento FunGramKB para actuar de apoyo a la hora de resolver de manera más exitosa este tipo de problemas del PLN.

6.4. LA METÁFORA Y LA METONIMIA

Otros dos fenómenos lingüísticos que entrañan una complejidad manifiesta, no sólo en relación a su tratamiento en el PLN, sino para la investigación lingüística en general, son la metáfora y la metonimia. El hecho de que hayamos elegido mostrar estos dos fenómenos dentro de una misma sección no es casual. Por el contrario, se debe a que en numerosas ocasiones la línea de separación a trazar entre ambos fenómenos no se puede hacer de forma nítida ya que, con más frecuencia de la esperada, encontramos que ambos se pueden posicionar, más que a un lado o a otro de una línea divisoria, a lo largo de un *continuum*. Ruiz de Mendoza Ibáñez (1999) indica que los extremos de este *continuum* son, por un lado, las proyecciones metafóricas con varias correspondencias y, por otro, las metonimias referenciales, mientras que en medio de ambas encontraríamos a las metáforas con una sola correspondencia y a los usos predicativos de la metonimia. Incluso, algunos autores han llegado a registrar fenómenos híbridos, como la denominada *metaftonimia* (Goossens, 1990), la cual más adelante vemos con mayor detalle.

Retornando a la metonimia, en primer lugar, ésta se define como un mecanismo conceptual mediante el cual queremos evocar una entidad meta a través de una entidad fuente, perteneciendo ambas al mismo dominio conceptual (cf. Ruiz de Mendoza Ibáñez

y Galera-Masegosa, 2011:5). Asimismo, encontramos dentro de la metonimia dos tipos diferentes de fenómenos, comúnmente designados “el todo por la parte” o “la parte por el todo”, los cuales Ruiz de Mendoza Ibáñez (2000) denomina con mayor precisión “*Target-in-source*” y “*Source-in-target*”, respectivamente.

Por otro lado, en relación a la definición de metáfora, la noción actual que poseemos al respecto dista mucho de la concepción clásica de la misma, que solía asignar el empleo de este recurso de manera casi exclusiva al lenguaje literario, desterrando su uso fuera del lenguaje común, de donde se desprendía, por ende, que la metáfora era entendida simplemente como un fenómeno lingüístico y no cognitivo (Lakoff, 1993). Sin embargo, la evolución de las teorías lingüísticas a lo largo del tiempo trajo consigo una nueva concepción de las expresiones metafóricas, ya no limitada exclusivamente al reino de las palabras, sino conectada muy de cerca con la mente y la comprensión del mundo que nos rodea, llegando a la conclusión de que “[...] *the locus of metaphor is thought, not language* [...]” (Lakoff, 1993:204), algo que la mayoría de autores contemporáneos parecen compartir. Como ejemplo, veamos las palabras de Peña Cervel que lo resumen:

“Cognitive theory has been able to show that, contrary to what was postulated by Reddy (1993) in connection with the conduit metaphor, meaning does not reside in the words themselves. Language is not regarded as the product of a separate mental ability within the brain but as part of the general cognitive processes which allow humans to conceptualize experience. These processes are called embodied understanding within the framework of Cognitive Linguistics (Johnson, 1987). [...] Meaning derives from our interaction with the world.”

(Peña Cervel, 2011:1).

De este modo, la definición de metáfora que podemos resumir como “*a cross-domain mapping in the conceptual system*” (Lakoff, 1993:203) se acerca mucho más a la concepción de dicho fenómeno en la cual basaremos nuestras disertaciones, ya que nuestro enfoque no queda anclado en metáforas no convencionales o literarias, sino que

trata de cubrir una gran parte de expresiones metafóricas de extendido uso en el lenguaje de un hablante nativo medio. Se trata, por tanto, de un recurso que nos ayuda, a través de la alusión a un dominio generalmente más concreto, a expresar o comprender un dominio más abstracto (cf. Ruiz de Mendoza Ibáñez y Galera-Masegosa, 2011).

Si bien resulta importante tener en cuenta los escritos de reconocidos autores en la materia como Lakoff (1987, 1993) o Langacker (1987, 1991), también debemos reconocer que existe otro tipo de definiciones que autores más cercanos al campo del PLN proporcionan acerca de los fenómenos metafóricos en el lenguaje, las cuales sirven de base más que adecuada para exponer las soluciones que FunGramKB puede aportar a la resolución de las expresiones metafóricas. De este modo, tomamos como punto de partida la definición que Shutova (2010) realiza de la metáfora: *“Metaphors arise when one concept is viewed in terms of the properties of the other. In other words it is based on similarity between the concepts.”* (Shutova, 2010:688). La mención a *“properties”* como rasgos definitorios o características de un concepto resulta muy conveniente para nuestro propósito, ya que esta misma nomenclatura la encontramos en el Onomasticón de FunGramKB.

Por tanto, si adaptamos esta definición de metáfora, entendida tanto en su concepción más amplia como fenómeno y como expresiones metafóricas más concretas⁸⁹, a los constructos conceptuales presentes en FunGramKB, podríamos decir que la metáfora en nuestra base de conocimiento supone una conexión entre conceptos a través de la cual los postulados de significado o el marco temático de uno de ellos son intercambiados con los de otro, basándose en una relación de similitud o semejanza que puede proceder de diversos orígenes, pero que en última instancia termina

⁸⁹ Lakoff (1993) traza la siguiente distinción de ‘expresión metafórica’ con respecto a la de metáfora: *“The term “metaphorical expression” refers to a linguistic expression (a word, phrase, or sentence) that is the surface realization of such a cross-domain mapping [...]”* (Lakoff, 1993: 203).

fundamentándose en nuestra experiencia del mundo. Para ilustrar esto, nos basaremos en un ejemplo de Wilks (1978) que muestra que existe una anomalía en los conceptos que se conectan:

(33) *My car drinks gasoline.* (Wilks, 1978:199)

La explicación proporcionada para este ejemplo se basa en que existe una violación en un contexto determinado de las preferencias de selección de un verbo, algo que el autor enuncia como un rasgo característico de las metáforas. En este caso, observamos que existe una transgresión de las preferencias de selección del verbo *drink* para adjudicarle un sujeto inanimado, ya que *drink* se aplica convencionalmente a un sujeto animado. Sin embargo, esta transgresión resulta muy interesante para el PLN, como a continuación explicamos.

Cuando se trata de diseñar métodos para la resolución de la metáfora, es útil basarse en modelos que contemplen dos fases muy diferenciadas, aunque complementarias: en primer lugar, la identificación o reconocimiento de la metáfora y, en segundo lugar, la interpretación de la misma. Para llevar a cabo la primera fase, Shutova (2010) sugiere un método ya incluido en el enfoque de Fass (1991) y que desde nuestra perspectiva en FunGramKB estimamos que también puede ser factible de realizar con nuestra base.

La fase inicial de reconocimiento o identificación de la metáfora puede basarse en uno de los rasgos inherentes a este fenómeno que mencionábamos con anterioridad, es decir, la violación de las preferencias de selección que lleva normalmente vinculadas por naturaleza semántica o conceptual un predicado. En este caso, la señal de alerta que un sistema de PLN puede emitir para indicar la localización de una metáfora estaría conectada con la detección de un cambio significativo en las preferencias de selección

asignadas a un predicado, lo que en FunGramKB detectaríamos mediante una alteración del marco temático (por ejemplo, una variación en los roles temáticos de un evento). Por tanto, retomando de nuevo el ejemplo (33), en FunGramKB encontraríamos la siguiente estructura para el subconcepto –DRINK, que pertenece al concepto +INGEST_00:

CONCEPTO	+INGEST_00 [-DRINK]
SUPERORDINADO(S)	+ABSORB_00
MARCO TEMÁTICO	(x1: +HUMAN_00 ^ +ANIMAL_00)Agent (x2: +LIQUID_00)Theme (x3: +THROAT_00)Location (x4)Origin (x5: +STOMACH_00)Goal
POSTULADO DE SIGNIFICADO	(e1: +ABSORB_00 (x1)Agent (x2)Theme (x6)Location (x4)Origin (x7: +MOUTH_00)Goal (f1: x6)Instrument (f2: (e2: +SWALLOW_00 (x1)Agent (x2)Theme (x3)Location (x7)Origin (x5)Goal))Purpose)
DESCRIPCIÓN	to take food, liquid, or some other substance into the body by swallowing

Tabla 22. El subconcepto –DRINK en la Ontología de FunGramKB.

El marco temático del subconcepto –DRINK nos indica que x1, es decir, el agente de la acción, tiene como preferencias de selección una entidad animal o humana, la cual ingiere una entidad líquida (x2) que pasa a través de la garganta (x3), desde una ubicación (x4) al estómago (x5). Por tanto, en el ejemplo “*My car drinks gasoline*”, distinguimos que los participantes x1, x3 y x5 han sufrido una transgresión, de tal forma que el agente (x1) ya no es humano o animal y que el medio (x3) no puede ser la garganta, ya que este concepto está vinculado únicamente a las entidades animales o humanas. Asimismo, la meta del líquido ingerido (x5) tampoco puede ser en este caso un concepto relativo a la anatomía animal o humana, ya que sólo dichas entidades poseen un estómago.

Por tanto, a la luz de este ejemplo, la base FunGramKB podría efectuar la detección inicial de una violación en las preferencias de selección de un predicado,

alertando así de un posible fenómeno metafórico. No obstante, sería preciso destacar que el propósito de esta tarea no debe ser la identificación de los errores en un texto, ya que en caso contrario, la máquina podría indicarnos simplemente que un texto es erróneo e incoherente y no que puede contener una expresión metafórica, el cual es nuestro propósito. Por tanto, el corpus de textos con el que alimentar a la máquina debe contener extractos de probada corrección y coherencia textual.

Hasta ahora, en el proceso de identificación de las metáforas que hemos ilustrado, no se precisaba la intervención del módulo Onomasticón de FunGramKB, ya que la información necesaria para detectar la existencia de una transgresión del marco temático ha sido posible gracias a la información contenida en la Ontología. Sin embargo, la inclusión de los contenidos del Onomasticón en este proceso lo enriquece de gran manera. Recordemos que la Ontología de FunGramKB sólo contiene información conceptual acerca de proto-microestructuras, por lo que la adición del Onomasticón a la base de conocimiento completa a ésta con información sobre entidades nombradas, las cuales pueden participar en el marco temático de un evento. Asimismo, el papel desambiguador que puede jugar la información acerca de las entidades nombradas es clave, como podemos observar en el siguiente ejemplo:

(34) *He bought a Ford.* (Peña Cervel, 2011:5)

En este ejemplo, encontramos un caso de la metonimia denominada “*Target-in-source*”, ya que estamos nombrando a una clase entera (i.e. la compañía automovilística Ford) por un elemento de dicha clase (i.e. un automóvil fabricado por la compañía Ford), lo que podemos resumir como una muestra de la metonimia COMPANY FOR PRODUCT (Peña Cervel, 2011). Gracias a que la compañía de automóviles Ford es una

entidad nombrada presente en el Onomasticón de FunGramKB, la máquina es capaz de reconocer qué es *Ford*.

Ahora bien, si combinamos los ejemplos (33) y (34), podemos avanzar un paso más en la capacidad de resolución de una combinación de metáfora y metonimia para la que el Onomasticón resulta de gran importancia. Observemos la siguiente oración:

(35) *Este Ford se bebe la gasolina de una manera pasmosa.*

En este ejemplo hemos combinado la metáfora presente en (33) y la metonimia en (34). Sin embargo, para poder determinar si la expresión en (35) se trata, efectivamente, de una metáfora mediante la cual expresamos que un coche *bebe* gasolina como si fuese una entidad animada, necesitamos primero conocer qué es la expresión *Ford* y qué tipo de entidad denota, ya que es posible que *Ford* sea una entidad animada y, por tanto, no estemos ante ningún fenómeno metonímico-metafórico. Puesto que *Ford* forma parte de las entidades del Onomasticón, a través de la información proyectada desde Wikipedia a través de DBpedia, la conexión “*Ford* → coche” que se establece en la metonimia COMPANY FOR PRODUCT sería posible de detectar por la máquina, gracias a la proyección que ilustramos en la figura que se muestra a continuación:

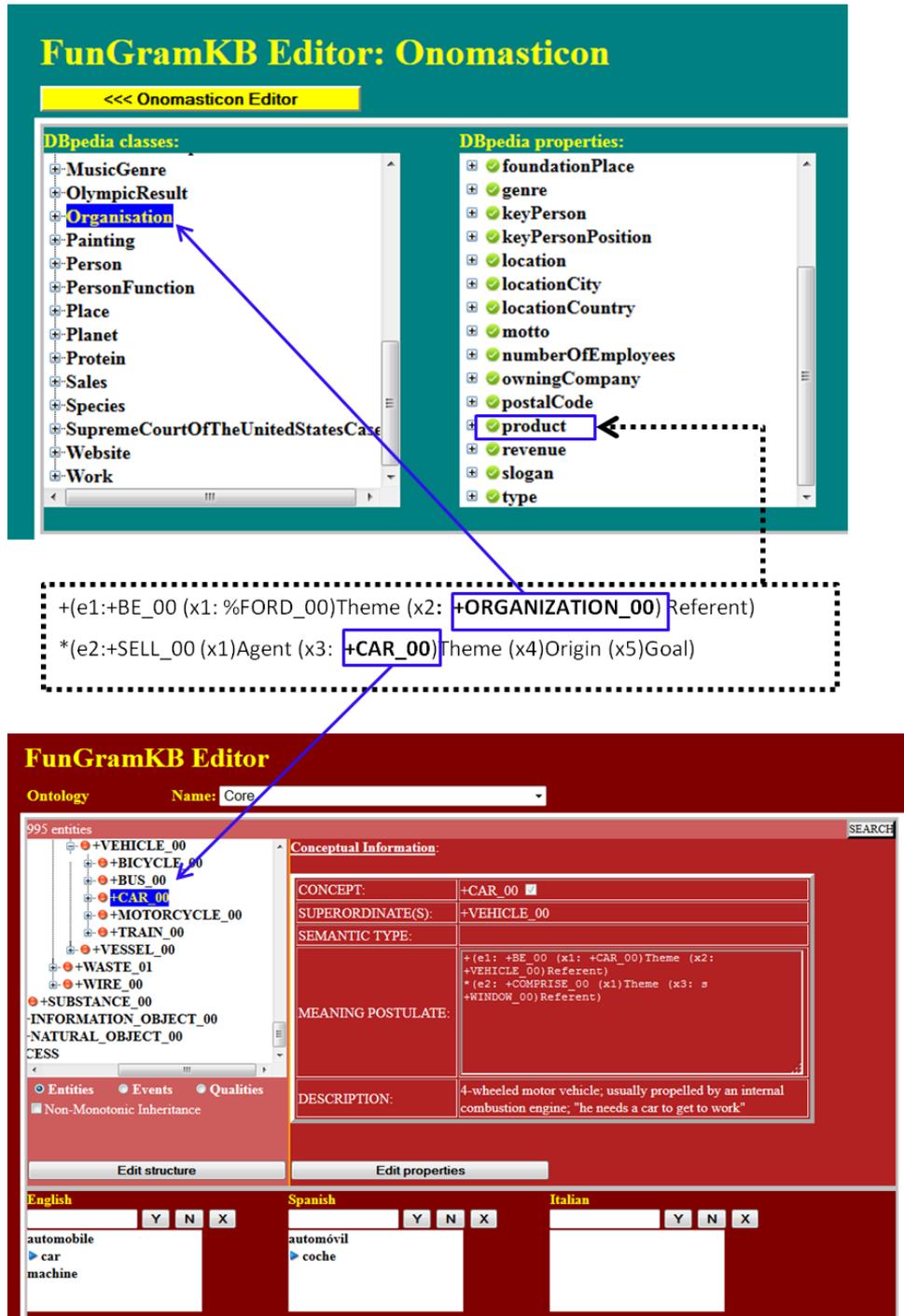


Figura 11. Conexión conceptual para la resolución metonímico-metafórica de la expresión “Ford”.

La Figura 11 se explica de la siguiente forma: en el recuadro central podemos ver el resultado de la proyección de los datos en el Onomasticon de FunGramKB de la regla para la propiedad *product* de la entidad %FORD_00, de tal forma que en este constructo conceptual está contenida la información que en lenguaje natural podemos

traducir como “*la entidad Ford es una organización, la cual vende automóviles*”. A través de esta proyección se establece el vínculo entre la entidad de la Ontología +ORGANIZATION_00 y la clase “Organisation” de DBpedia. Asimismo, observamos que también existe una conexión entre el concepto de la Ontología +CAR_00 y la propiedad *product*, ya que dicho concepto constituye el valor de esta propiedad. Por tanto, gracias a la proyección efectuada es posible que la máquina efectúe una conexión entre la entidad *Ford*, la clase “Organisation” y el producto que comercializa dicha entidad (i.e. +CAR_00), de tal forma que se pueda resolver la combinación metafórico-metonímica planteada en el ejemplo (35).

De este modo, observamos la ventaja que puede suponer la introducción de conocimiento cultural en una herramienta de PLN que pretenda de alguna manera resolver o, al menos, ser capaz de identificar con éxito fenómenos metonímico-metafóricos en los que se combina el lenguaje natural con la inclusión de entidades nombradas que aportan conocimiento cultural.

6.4.1. LA METAFTONIMIA

Un fenómeno en el que se combinan la metáfora y la metonimia es la *metaftonimia*. El término “*metaphonymy*” fue acuñado inicialmente por Goossens (1990), si bien con objeto de utilizarlo en nuestro trabajo adoptamos el enfoque de Brdar y Brdar-Szabó (2007), quienes se centran en los nombres propios y en la figura del parangón, así como el enfoque de Pang (2010), que muestra los efectos del uso espurio de los nombres propios que termina dando origen a la eponimia y al uso de la construcción “*X is the Y of Z*” como la que encontramos, por ejemplo, en la expresión “*Singapore is the Manhattan of Asia.*” (Pang, 2010:1321).

Acerca de la eponimia, ésta tiene una gran base en las convenciones culturales compartidas por un grupo social o una civilización sobre un personaje o nombre propio,

el cual pasa a designar un concepto vinculado a él mismo. Tanto es así, que podría decirse que las construcciones eponímicas cumplen una función unificadora a nivel cultural: *“Eponymic constructions are linguistic triggers to analogical conceptualization, i.e., “instructions” to the interlocutors to set up networks of analogy spaces.”* (Pang, 2010:1329). Por ejemplo, un titular periodístico como *“El Supremo valida que la Administración pueda cerrar webs como contempla la ley Sinde⁹⁰”* sería una muestra de cómo una entidad nombrada pasa a constituir un referente cultural en un momento y lugar determinados, la *“ley Sinde⁹¹”*.

Además de mencionar la eponimia, resulta interesante dedicar nuestra atención a otra construcción vinculada con la figura del parangón, la cual denominamos *“X is the Y of Z”*. Esta construcción, así como las variantes *“Det + X_{personal name} + of Y”* o *“Y’s X_{personal name}”*, donde *“Y”* denota una ubicación o un periodo temporal, las cuales encontramos en Brdar y Brdar-Szabó (2007), se utilizan con el propósito de ilustrar que una entidad nombrada *“Y”* posee, en unas circunstancias o lugar establecidos, unas características similares, por analogía, a las que otra entidad nombrada *“X”* posee o ha poseído a lo largo del tiempo, siendo *“X”* un referente cultural o social enraizado en el conocimiento cultural del mundo que ambos interlocutores deben compartir. En palabras de Brdar y Brdar-Szabó: *“Paragons as metonymic models, just like generators, ideal cases, social stereotypes, etc., are said to operate on the basis of the same single metonymic process, which is called CATEGORY FOR DEFINING PROPERTY by Kövecses and Radden [...].”* (Brdar y Brdar-Szabó, 2007:131).

⁹⁰ Obtenido de <http://www.rtve.es/noticias/20130620/supremo-valida-administracion-pueda-cerrar-webs-como-contempla-ley-sinde/693442.shtml>, fecha de consulta 10/02/2014.

⁹¹ Se refiere a la Disposición final cuadragésima tercera de la Ley 2/2011 de 4 de marzo (Ley de Economía Sostenible), promovida por la entonces Ministra de Cultura Ángeles González-Sinde, en la que se regula el funcionamiento de la Comisión de Propiedad Intelectual en España, en relación a la vulneración de derechos de autor en páginas web.

Con objeto de desarrollar esta visión de manera más ilustrativa, observemos el siguiente ejemplo:

(36)

Toni Morrison lives in the ‘Champions League’ of literature. She is the Zidane of language, beguiling, intoxicating and deadly in front of goal.

(Brdar y Brdar-Szabó, 2007:126).

El uso de la expresión “*the Zidane of language*” en (36) presupone que ambos, emisor y receptor del mensaje, comparten un acervo cultural común en el cual se incluyen las capacidades de la entidad nombrada *Zidane*, así como que el ámbito de actuación y experiencia de dicha entidad no es el lenguaje. Sin embargo, además de hallar la metonimia CATEGORY FOR DEFINING PROPERTY, también encontramos una correspondencia metafórica que se explicaría de este modo:

“First of all, Zidane in the examples [...] must be interpreted as ‘Zidane the footballer,’ i.e. we have a WHOLE FOR PART mapping taking place here. The whole of our encyclopedic knowledge that we have about the person labelled by the proper name referring to a particular real person, i.e. Zidane, must constitute a domain of its own.”

(Brdar y Brdar-Szabó, 2007:132).

Por tanto, podemos considerar que nos encontramos frente a uno de los resultados que ofrece el uso figurativo de los nombres propios, el cual podríamos denominar *metaftonímico*, ya que se trata de un fenómeno en el que se produce una combinación simultánea de correspondencias metafóricas y metonímicas. La correspondencia se explica de la siguiente manera: por un lado, las cualidades particulares de la entidad nombrada *Zidane* (i.e. la excelencia en el campo futbolístico) sufren un *foregrounding*, es decir, son destacadas de manera prominente como rasgo representativo de la entidad, por lo que ésta pasa a convertirse en un dominio en sí misma, como indicaban Brdar y Brdar-Szabó en la cita arriba mencionada (2007:132). Podríamos decir, por tanto, que en el discurso mostrado en (36) la entidad *Zidane* sufre

una metaforización, dado que se convierte en un dominio origen para expresar otro dominio (i.e. la excelencia en el campo literario). Sin embargo, por otro lado, también se puede considerar que la entidad *Zidane* se ha transformado en una expresión metonímica, ya que mediante la alusión a *Zidane* no nos estamos refiriendo a otras cualidades del mismo que no han sufrido *foregrounding*, como sería una alusión a *Zidane la persona francesa* o a *Zidane el padre*, sino que nos referimos solamente a *Zidane el futbolista*, siendo esta característica una parte del todo que es la persona, como resumen Brdar y Brdar-Szabó:

“First, within this domain intersection, comprising all the information about Zidane that is relevant to conceptualizing him as a footballer, we suppress the majority of elements that are shared by all football players, and highlight those elements that make him different, even search for his unique traits, such as precise passes, scoring from free shots, etc.

In other words, we activate the BEARER OF PROPERTY FOR CHARACTERISTIC PROPERTY metonymy that brings us closer to the “X endowed with immense talent/skill as far as Y is concerned” meaning [...].”

(Brdar y Brdar-Szabó, 2007:135).

Esta correspondencia es expresada de manera gráfica en la figura que mostramos a continuación:

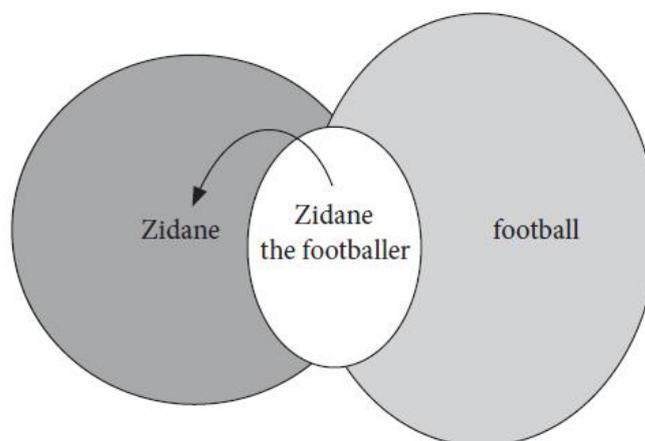


Figura 12. Intersección de los dominios como meta para la metonimia ENTIDAD POR ZONA ACTIVA (Brdar y Brdar-Szabó, 2007:132).

Un nuevo ejemplo de estas correspondencias metafonémicas en las que se integra la metonimia BEARER OF PROPERTY FOR CHARACTERISTIC PROPERTY lo podemos encontrar en la siguiente expresión:

(37) *Albacete, Nueva York de La Mancha*⁹²

En este caso, el parangón se produce al observar el poeta, a su paso en un tren a comienzos del siglo XX, la silueta recortada de la ciudad de Albacete en medio de una inmensa llanura, siendo iluminada por la todavía recién desarrollada luz eléctrica y en medio de un frío páramo. En ese momento, el poeta efectúa la comparación de esta imagen con la silueta de la ciudad de Nueva York emergiendo desde la lejanía, lo que resulta una bella y evocadora visión de la ciudad castellano-manchega.

Sin embargo, la resolución exitosa de este parangón puede tornarse complicada incluso para un hablante nativo de español, ya que este tipo de construcciones suele implicar una carga emotiva o personal del emisor del mensaje que, en numerosas ocasiones, puede no ser correctamente interpretada o compartida por el receptor del mismo, a menos que se trate de una referencia cultural de contrastada popularidad. El problema subyace, principalmente, en el desconocimiento de las propiedades que sufren el efecto del *foregrounding* de la entidad que se toma de referencia. Por ejemplo, la expresión mostrada en (37) puede suscitar dudas acerca de su resolución por parte de un receptor del mensaje que no conozca la situación en la que ésta se ha desarrollado o, simplemente, que ignore qué características de la ciudad de Nueva York, de las múltiples opciones posibles, se pretenden destacar en la expresión enunciada (p. ej. la altura de los edificios, las dimensiones de la ciudad, la mezcla de culturas, la representatividad de la nación a la que pertenece, etc.).

⁹² Título de un poema de José Martínez Ruiz “Azorín” (1873-1967).

Por tanto, si para un humano puede resultar complicado el identificar de manera correcta las correspondencias que establece este tipo de parangones, la manera de resolverlas basándose en la popularidad u homogeneidad del conocimiento cultural puede tornarse, nuevamente, una utopía para las herramientas de PLN. Sin embargo, algunos recursos computacionales pueden resultar de gran ayuda para su resolución, a pesar de que, a priori, la comprensión de este tipo de construcciones por parte de la máquina pueda parecer incluso más compleja. Tomemos el siguiente texto como ejemplo:

- (38) “[. . .] while ***I’m no Fred Astaire***, I’m a damn fine wedding dancer [...].”
(Pang, 2010:1323).

En el ejemplo (38), a pesar de no encontrar la estructura típica “*Det + X_{personal} name⁺ of Y*” ni ninguna de sus variantes, lo que sí detectamos es que se hace uso de una entidad nombrada, *Fred Astaire*, en la que nuevamente aparece una metonimia del tipo BEARER OF PROPERTY FOR CHARACTERISTIC PROPERTY. Si bien la resolución de este tipo de metonimia (o *metaftonimia*) parece demasiado compleja como para ser solventada con éxito por parte de la máquina, en el caso de utilizar la base de conocimiento FunGramKB podemos vislumbrar la posibilidad de ser resuelta con éxito, de nuevo gracias a la información contenida en el Onomasticón.

Puesto que la entidad nombrada *Fred Astaire* es una persona, ésta aparecerá dentro del inventario de entidades que pertenecen a la clase “Person” de DBpedia contenida en el Onomasticón de FunGramKB. Como hemos mencionado con anterioridad, cada clase del Onomasticón posee una lista de propiedades que se corresponden con los epígrafes de la ficha informativa de Wikipedia que describe a dicha entidad. En el caso de las entidades catalogadas como “Person”, encontramos una propiedad de gran interés para nuestro propósito, denominada *occupation*. Dentro de

esta propiedad, la información que se nos proporciona constituye un excelente indicador acerca de la profesión por la que se conoce a una entidad nombrada perteneciente a la clase “Person”, así que es a través de dicha propiedad como podemos efectuar la conexión. En el caso concreto de la entidad *Fred Astaire* mencionada en (38), encontramos que Wikipedia, dentro de su ficha informativa, nos proporciona información clave relacionada con la profesión de dicha entidad, como mostramos en la proyección que podemos ver a continuación:

(39)

`+(e1:+BE_00 (x1: %FRED_ASTAIRE_00)Theme (x2:+HUMAN_00)Referent)`

`*(e2: +DO_00 (x1)Theme (x3: +ACTOR_00 & $DANCER_00 & $SINGER_00 & $CHOREOGRAPHER_00 & +MUSICIAN_00) Referent (f1:(e3:+BE_00 (x3)Theme (x4:+JOB_00)Referent))Condition)`

Fred Astaire es una persona cuya profesión es la de actor, bailarín, cantante, coreógrafo y músico percusionista.

En esta proyección podemos observar que la ocupación que se atribuye a la entidad *Fred Astaire* es la de actor, bailarín, cantante, coreógrafo y percusionista. Por tanto, si efectuamos la sustitución de la expresión original *Fred Astaire* en el ejemplo (38), obtendremos la siguiente expresión:

(40) [...] while **I’m no actor, dancer, singer, choreographer or percussionist**, I’m a damn fine wedding dancer [...].

A pesar de que la oración en (40) pueda resultar cuanto menos curiosa a nivel estilístico o pragmático, es indudable que para la máquina esta frase posee coherencia y cohesión textual y, lo que es más importante, constituye un conjunto de conceptos comprensibles por ella, dado que los conceptos que denotan estas lexicalizaciones se pueden encontrar dentro de la Ontología de FunGramKB.

Sin embargo, la principal problemática que emerge en estos casos de *metaftonimia* tiene que ver con el ejercicio de *foregrounding* que debe ser realizado previamente para resolver con éxito este tipo de ejemplos. En concreto, sería deseable contar con un algoritmo que permitiese la realización del proceso de *foregrounding* de manera automática por la máquina. Sin embargo, en muchas ocasiones este *foregrounding* resulta complicado de realizar correctamente incluso por un humano (i.e. como sucede en el ejemplo (37) visto con anterioridad), razón por la que la máquina necesitaría recibir información adicional para poder resolverlo con éxito. Esta situación nos conduce a pensar que sería necesario, de cualquier forma, pasar por una fase asistida antes de poder ejecutar la fase automática, algo que no nos resulta ajeno a la vista de la metodología de población semi-automática del Onomasticón. Una posible solución para ayudar a que la máquina efectúe el proceso de *foregrounding* podría ser la aportación previa, de manera manual o asistida, de un peso o prioridad mayor a aquellas propiedades que consideramos más prominentes en cada tipo de entidad. Esta ponderación podría aplicarse sobre las propiedades de la ficha informativa de las entidades en Wikipedia, de tal forma que ciertas propiedades (p. ej. *occupation* o *knownFor* para una persona) fueran asignadas un peso mayor en función de su representatividad de la clase de entidades a las que se vinculan. La manera de otorgar esta prominencia podría basarse en las nociones de la *lógica difusa* (Zadeh, 1965; 1968), de tal forma que en una escala de 0 a 1 se confiriese a cada propiedad de la entidad un valor determinado, de mayor a menor representatividad. Una vez realizado este ejercicio con las propiedades de una entidad, podría realizarse posteriormente un proceso similar con los valores de cada una de ellas. Así, por ejemplo, en la proyección que observamos en (39) podríamos situar en el *continuum* de prominencia más cerca del número 1 al valor \$DANCER_00, seguido de +ACTOR_00 y así sucesivamente, debido

a que culturalmente la entidad *Fred Astaire* parece estar más reconocida como bailarín o como actor que como músico percusionista. No obstante, este ejercicio plantearía un nuevo dilema, el cual emerge del hecho de que la asignación de esta prominencia deba venir determinada por parte de un humano, razón por la podemos argüir que dicha tarea puede estar impregnada de una significativa carga de subjetividad o incluso de arbitrariedad. El problema reside, por tanto, no sólo en el hecho de tener que conferir un valor determinado a la prominencia de unas propiedades en favor de otras, sino en ser capaces de elaborar un algoritmo que permita que este proceso de *foregrounding* se realice de manera automática, lo que supondría una interesante línea de investigación futura en este ámbito.

No obstante, en conclusión, a la luz de los diversos ejemplos mostrados para cada uno de los problemas lingüísticos del PLN, encontramos que en el presente capítulo hemos dado cumplimiento al tercero de los objetivos enunciados al comienzo de este trabajo, puesto que hemos mostrado cómo la inclusión de conocimiento cultural a través del Onomasticón posibilita la ampliación del potencial de dicho modelo de conocimiento, con objeto de ayudar a desambiguar fenómenos lingüísticos que pueden resultar problemáticos para el PLN como son la referencia y correferencia, la anáfora, la metáfora, la metonimia y la *metaftonimia*.

A continuación, en el capítulo final de este trabajo presentamos una recapitulación de los temas desarrollados y de las conclusiones que de éstos podemos obtener. Asimismo, evaluamos el cumplimiento de los objetivos enunciados al comienzo de esta tesis y planteamos una serie de cuestiones relacionadas con nuestra investigación, junto a las futuras posibilidades que se abren en la misma al término de la presente tesis doctoral.

CAPÍTULO VII

CONCLUSIONES

7.1. RESUMEN DE LA PRESENTE TESIS DOCTORAL

La presente tesis doctoral comenzó con un planteamiento triple en relación a sus objetivos. Por una parte, el objetivo primario consistía en desarrollar una metodología de población semi-automática del módulo denominado Onomasticón de FunGramKB, una base de conocimiento multilingüe y multipropósito basada en la semántica profunda, la cual ha sido analizada con detalle, así como los resultados obtenidos tras la aplicación de dicha metodología y las características particulares que éstos presentan. Por otra parte, el segundo objetivo consistía en averiguar, a través del análisis de los resultados obtenidos, si éstos presentaban ciertas características similares, tanto a nivel formal como a nivel semántico, a partir de lo cual se pudiera confirmar o rebatir la siguiente hipótesis: una similitud formal en las estructuras y construcciones obtenidas puede suponer una motivación semántica compartida. Finalmente, el tercero de nuestros objetivos pretendía esclarecer el posible impacto que nuestro trabajo en el Onomasticón de FunGramKB podía tener sobre la resolución de problemas lingüísticos del PLN.

Así pues, al inicio de este trabajo hemos comenzado por describir el marco en el que se encuadra nuestra investigación, con objeto de contextualizarla en el estado de la

técnica actual, efectuando una descripción de lo que constituye la migración de datos y el aprovechamiento y reutilización de los recursos ontológicos.

A continuación, hemos mostrado una detallada visión del tipo de modelos de conocimiento vinculados a nuestra investigación, esto es, ontologías y bases de conocimiento, junto a las diferentes características que pueden definir las y cómo se diferencian entre ellas. Asimismo, hemos mencionado diversos proyectos relacionados con la proyección de datos y la migración de los recursos ontológicos, especialmente en conexión con varias iniciativas ligadas a la Web Semántica (p. ej. el proyecto *Linked Data Web*), además de otros proyectos similares al nuestro, si bien enmarcados en diferentes instrumentos y metodologías (p. ej. Serene, Swoogle, DEQA, SWSE, Cyc). Seguidamente, hemos aprovechado para enumerar los diferentes enfoques que pueden ser utilizados en la creación de modelos ontológicos dependiendo del mayor o menor grado de inclusión de conocimiento de carácter semántico o de carácter relacional, lo que nos aporta tres enfoques diferentes (i.e. enfoque superficial, enfoque intermedio y enfoque profundo), así como la decisión de contar con la implicación de una teoría lingüística sólida que respalde el instrumento creado. Una vez efectuada esta descripción de ontologías y bases de conocimiento, hemos trasladado nuestra discusión a la base FunGramKB.

Además de desarrollar de manera completa la arquitectura y contenido de la base de conocimiento FunGramKB, se ha procedido a realizar un retrato más pormenorizado del módulo Onomasticón de dicha base, el cual contiene conocimiento cultural a través de la presencia de entidades nombradas. Igualmente, a continuación hemos analizado otros proyectos – YAGO, YAGO2 y DBpedia – que reconocen el enorme valor informativo acerca de entidades nombradas que el portal Wikipedia puede ofrecer, así como la utilidad de organizarlo dentro de una ontología o una base de conocimiento.

Para poder efectuar la migración de este conocimiento desde Wikipedia a FunGramKB, el Onomasticón se nutre de la base de conocimiento DBpedia, lo que supone un ejercicio de optimización y reutilización de recursos ya existentes.

Con posterioridad a la enumeración de los beneficios que la interacción y reutilización de estos recursos ontológicos puede conllevar para FunGramKB en particular y para cualquier herramienta de PLN en general, el foco de nuestra discusión se ha trasladado a comenzar a acometer nuestro primer objetivo, mediante la descripción pormenorizada de la metodología de población del Onomasticón. Este proceso se realiza de manera semi-automática a través de tres fases, siendo la primera de ellas la fase asistida (fase de construcción) y siendo las dos restantes (fase de proyección y fase de actualización) las fases automáticas. Durante la descripción de esta metodología hemos prestado atención especial a los condicionantes y situaciones decisivas a lo largo de la implementación de la misma, especialmente durante la fase asistida del proceso, que comprende el protocolo de elaboración de *reglas* (i.e. plantillas en lenguaje COREL) que permiten la proyección de datos desde la base de conocimiento DBpedia hasta el Onomasticón de FunGramKB.

Posteriormente, una vez especificada la metodología de población del módulo Onomasticón de FunGramKB, nuestro enfoque se ha dirigido hacia el segundo de nuestros objetivos, comenzando por un análisis pormenorizado de los resultados obtenidos tras la implementación de dicha metodología. Este análisis, realizado tanto desde una perspectiva semántica como desde otra más formal, ha arrojado dos tipologías diferentes para clasificar las plantillas o reglas elaboradas para el presente trabajo. Asimismo, la observación minuciosa de estas tipologías y las conclusiones que de ellas se podían desprender nos han aportado una confirmación de la hipótesis enunciada en el segundo objetivo planteado al principio de este trabajo: efectivamente,

se ha constatado que una estructura formal compartida puede significar no sólo una motivación semántica común, sino que es posible, incluso, vincular los diferentes tipos de construcciones enumerados en la tipología formal con los diferentes tipos de propiedades contempladas en la clasificación semántica.

Sin embargo, a pesar de haber confirmado nuestra hipótesis inicial, el presente trabajo no podía dar por concluidos sus objetivos en este punto, sino que debía continuar más allá, debido a que todavía quedaba por resolver un tercer objetivo que respondía a una motivación ulterior, basada en la provisión de herramientas y modelos de conocimiento de semántica profunda para su uso en PLN y las ventajas que de ello pudieran obtenerse. Así pues, en cumplimiento de nuestro tercer objetivo, hemos realizado una presentación del impacto que la población del Onomasticón tiene en relación a cuestiones lingüísticas, efectuando una muestra de cómo esta población es capaz de enriquecer la base de conocimiento FunGramKB. Gracias a la inclusión de conocimiento enciclopédico o cultural en FunGramKB observamos notables progresos en la resolución de problemas lingüísticos que acucian al PLN como son la referencia y la correferencia, la anáfora, la metáfora, la metonimia y la *metaftonimia*.

Por tanto, una vez que hemos efectuado el resumen del trabajo desarrollado en la presente tesis doctoral, así como de la forma en que sus objetivos han sido conseguidos, cabe plantearse cuáles son las conclusiones que se pueden derivar del estudio completo. Igualmente, consideramos interesante mencionar las implicaciones de nuestro trabajo con respecto a diversas cuestiones y asuntos transversales con los que se puede establecer una conexión.

7.2. CUESTIONES TRANSVERSALES DE LA MIGRACIÓN DE DATOS

En el Capítulo II de la presente tesis doctoral comenzábamos mencionando la iniciativa *Linked Data Web*, a través de la cual podría evolucionar la WWW tal como se había implementado hasta el momento, para pasar a transformarse en un medio real de interacción entre usuarios y no sólo una herramienta de lectura. Por tanto, observábamos que numerosos proyectos y herramientas hoy en día se han sumado a dicha iniciativa del W3C, incluyendo el proyecto DBpedia que sirve como apoyo principal a la población del Onomasticón de FunGramKB.

Por otro lado, esta interconexión y disponibilidad de grandes repositorios de datos se vincula cada vez más con el término *Big Data*, es decir, con ingentes cantidades de información que crecen de manera exponencial y que aglutinan datos recabados de múltiples fuentes, entre las que se hallan, por supuesto, aquellas que encontramos en la *Linked Data Web*, en especial los buscadores web más populares como Google, Yahoo! o Bing, entre otros, así como agencias gubernamentales y otras instituciones.

El desarrollo e implementación de proyectos de migración de conocimiento como la *Linked Data Web* y de recopilación de *Big Data* presentan, tanto a nivel computacional como lingüístico, un importante potencial para el desarrollo de la Web Semántica y de otras herramientas de PLN. Mediante la interconexión de los datos obtenidos en los proyectos mencionados, la Web Semántica puede verse enormemente enriquecida, siempre y cuando el contenido a proyectar sea tratado adecuadamente y convertido en conocimiento semántico, lo que supondría enormes ventajas en términos de utilidad para el usuario. Esto podría suponer desde una interacción más inteligente entre humano y máquina o entre máquinas, pasando por una reducción considerable del tiempo invertido en búsquedas web, hasta los ejemplos de domótica que hemos

mencionado en el Capítulo II (“[...] *It is not hard to imagine your Web-enabled microwave oven consulting the frozen-food manufacturer's Web site for optimal cooking parameters.*” Berners-Lee *et al.*, 2001:16).

Si bien nuestro trabajo como lingüistas computacionales o ingenieros del conocimiento centra sus esfuerzos de mayor manera sobre el PLN y los desarrollos de éste y, en mucha menor medida, sobre cuestiones de minería de datos o de recopilación de los mismos, estimamos que existe una cuestión emergente en las herramientas computacionales que no podemos ignorar. Esta cuestión es la privacidad de los datos que los usuarios proporcionan y que pueden ser migrados o tratados más allá de su origen, lo que representa una problemática que creemos que debería permear a través de toda investigación relacionada con la WWW, debido a la interconexión natural que observamos surgir en este punto entre computación y sociología. Si bien en el estado actual de nuestra investigación parece que esta materia no presenta todavía un problema real, creemos que no debemos obviar la posibilidad de que, en un futuro, las herramientas de PLN alcancen un grado de madurez tal que consiga entender mucho más de lo que nos imaginamos con respecto a la comunicación humana.

Para ilustrar los interrogantes que pueden plantearse en relación con este asunto, concluimos esta sección con una cita que resume de manera inquisitiva la problemática que surge a la hora de conciliar el equilibrio entre una utilidad ampliada de las herramientas digitales actuales (i.e. la Web Semántica o herramientas de minería de datos relacionadas con los *Big Data*) y la privacidad de los usuarios:

“Will large scale search data help us create better tools, services, and public goods? Or will it usher in a new wave of privacy incursions and invasive marketing? Will data analytics help us understand online communities and political movements? Or will analytics be used to track protesters and suppress speech? Will large quantities of

data transform how we study human communication and culture, or narrow the palette of research options and alter what ‘research’ means?”

(Boyd y Crawford, 2012:663)

7.3. CONCLUSIONES, APLICACIONES Y FUTURA INVESTIGACIÓN

Retomando nuevamente los temas centrales de la presente tesis doctoral, clausuramos este trabajo mediante la enumeración de las conclusiones generales que de éste se desprenden, así como de las aplicaciones para el PLN que pueden verse beneficiadas a través de nuestra investigación. Finalmente, describimos las futuras líneas de investigación que surgen al término de esta tarea.

En primer lugar, una conclusión genérica que podemos trazar está relacionada con el propósito de nuestro trabajo. Esto es, debemos mantener la perspectiva de que el principal interés que tenemos con la base de conocimiento FunGramKB es de carácter lingüístico-conceptual y se centra, principalmente, en efectuar mejoras a los sistemas de PLN, por lo que no debemos dejarnos cegar por los *Big Data* o por herramientas de minería de datos *sin sentido*⁹³: una abundancia de datos cuantitativos no significa, necesariamente, una mejora de nuestra herramienta a nivel cualitativo. Este pensamiento nos lleva a mencionar otra de las conclusiones que podemos efectuar del presente trabajo, la cual está relacionada con la principal herramienta que hemos utilizado para posibilitar la población del Onomasticón, i.e. la base de datos DBpedia.

Si bien hemos observado que la base DBpedia puede resultar un recurso extremadamente útil para la población semi-automática del Onomasticón de FunGramKB, existen también ciertos detalles con respecto a ella que precisarían de un mayor perfeccionamiento para optimizar la explotación de dicho recurso. DBpedia adolece particularmente de que la información disponible acerca de cada propiedad de

⁹³ Entendiendo la expresión *sin sentido* como sin sentido conceptual o semántico.

las entidades puede llegar a resultar inconsistente o incompleta, por lo que sería deseable contar con una mejora que permitiese que la información proporcionada acerca de cada propiedad fuese homogénea. Constituiría un importante avance el hecho de que para cada una de las propiedades se aportara una descripción de la misma de manera desarrollada, una unidad de medida y la clase a la que pertenece, entre otros aspectos relevantes que completaran su descripción, con el fin de que estos datos pudieran ser reutilizados y trasladados a otras herramientas de una manera más óptima, en especial al Onomasticón de FunGramKB. Obviamente, intuimos que esta posible organización, en ocasiones caótica, puede venir heredada de Wikipedia, principal recurso que nutre DBpedia y cuya heterogénea estructura no podemos ignorar.

Asimismo, este carácter complejo de DBpedia se refleja también en la propia organización de las clases de entidades presentes en ella, lo que parece indicar que más que una jerarquía de entidades, DBpedia se podría interpretar como una jerarquía de atributos, quizás más orientada a tareas de minería de datos que a la resolución de problemas conectados con el PLN. Además, observamos que, en ciertos casos, existe una jerarquía conceptual algo imprecisa, ya que a veces encontramos que, al amparo de dicha jerarquía, una entidad determinada podría adscribirse a la vez a dos clases jerárquicamente conectadas, razón por la cual es siempre necesario efectuar un filtro previo por parte del ingeniero del conocimiento aplicando el *sentido común*. A este respecto, consideramos que otra de las conclusiones que podemos trazar es que, debido a la permanente actualización de las fuentes, sería necesario efectuar una revisión de las reglas elaboradas y de las clases de DBpedia incorporadas en el DBpedia Mapper de FunGramKB. Igualmente, esta tarea de revisión también debería realizarse en caso de llevar a cabo una actualización del propio DBpedia Mapper de FunGramKB, como aplicación en su conjunto, a la última versión de DBpedia, puesto que de manera

habitual en FunGramKB se trabaja con una versión cristalizada de dicha base de conocimiento. En concreto, actualmente se trabaja con la versión 3.4 del DBpedia Mapper.

Por otro lado, en relación a las posibles aplicaciones para tareas de PLN que podemos obtener de FunGramKB, además de la migración de datos para la Web Semántica, cabe mencionar ciertas tareas de uso eminentemente práctico, como puede ser la creación de diccionarios o recursos lexicográficos digitales que permitan la búsqueda conceptual de términos. Otras opciones de aplicación están relacionadas con la traducción automática, algo que ya se ha probado con éxito, por ejemplo, en el enriquecimiento de UniArab, una herramienta de traducción automática de árabe estándar moderno a inglés, mediante la inserción de contenido procedente de FunGramKB (Periñán Pascual y Mairal Usón, 2010a). Además, otras aplicaciones de FunGramKB podrían llevarse a cabo también en el ámbito de la comunicación y la mercadotecnia, concretamente a la hora de realizar estudios de mercado. Por ejemplo, una interpretación semántica de las opiniones subjetivas vertidas por los usuarios de un producto en foros de opinión, lo que se denomina *sentiment analysis*, podría proporcionar a los fabricantes una amplia visión de qué es lo que gusta de su producto y qué ventajas o inconvenientes posee con respecto a los de otros competidores, sin necesidad de tener que leer una a una las opiniones o textos escritos por los usuarios. Gracias a la inclusión del conocimiento cultural del Onomasticón de FunGramKB, esta información podría contener, además, toda una identificación completa de las marcas, organizaciones y empresas presentes en los comentarios, puesto que se trata de entidades nombradas. En este sentido, destacamos el trabajo preliminar de Fernández Fernández y Sacramento Lechado (2012), cuyo análisis de sentimiento basado en el modelo de FunGramKB se centra en las diversas opiniones de consumidores acerca de

productos del mundo del vino. En última instancia, y también en conexión con el mundo de la organización corporativa, podemos mencionar otra posible aplicación del Onomasticón de FunGramKB, de tal forma que éste contenga conocimiento no sólo acerca de entidades nombradas culturalmente conocidas (i.e. conocimiento enciclopédico), sino también acerca de otro tipo de entidades nombradas de un ámbito concreto (p. ej. el personal de una empresa u organización, los pacientes de una clínica, los alumnos de un centro escolar, etc.), el cual es introducido de forma asistida, de tal forma que FunGramKB se transforme en un sistema de PLN capaz de solventar consultas acerca de personas o usuarios concretos identificados. Como posibles ejemplos de esta potencial aplicación de FunGramKB podríamos señalar la gestión de historiales médicos de los pacientes de una clínica, la implementación de una interfaz de comunicación entre un sistema determinado de una empresa y los empleados de la misma, la elaboración de una plataforma de gestión de expedientes académicos de una institución académica, etc. y en general, la transformación de cualquier plataforma digital que requiera la interacción del conocimiento acerca de personas o entidades nombradas que podría albergar el Onomasticón, con conocimiento semántico como el que alberga la Ontología y con conocimiento enciclopédico como el que se obtiene a través de DBpedia y que puebla el Onomasticón de FunGramKB.

Finalmente, en relación a las líneas de investigación que se abren al término del presente trabajo, podemos mencionar la actualización de las reglas del Onomasticón de FunGramKB ya creadas, así como una ampliación de contenidos de dicho módulo que incluya reglas para las propiedades de aquellas clases actualmente no pobladas, debido a su carácter minoritario o poco relevante hasta el momento. Asimismo, efectuar una ampliación de la base de conocimiento FunGramKB a través de la creación y perfeccionamiento de subontologías satélites basadas en terminologías específicas como

puede ser, por ejemplo, la terminología legal –actualmente en proceso– u otro tipo de terminología científico-técnica (p. ej., basada en la clase “Species” de DBpedia, descartada para este trabajo) pueden constituir los próximos desafíos que nos ofrece esta herramienta en el futuro más próximo.

BIBLIOGRAFÍA

- Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26 (11), 832-843.
- Allen, J.F. y Ferguson, G. (1994). Actions and events in temporal logic. *Journal of Logic and Computation*, 4 (5), 531- 579.
- Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R. e Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. En Aberer *et al.* (Eds.), *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007*. Berlin / Heidelberg: Springer.
- Auer, S. y Lehmann, J. (2010). Creating knowledge out of interlinked data. *Semantic Web*, Volume 1, Issue 1, 97-104.
- Baillie, C., Edwards, P. and Pignotti, E. (2012). Quality Reasoning in the Semantic Web. En *Proceedings of the International Semantic Web Conference 2012*, Boston, USA, November 2012, 383-390.
- Bateman, J.A. (1991). The theoretical status of ontologies in natural language processing. En S. Preuss y B. Schmitz (Eds.), *Text Representation and Domain Modelling - Ideas from Linguistics and AI, KIT-Report 97* (pp. 50-99). Berlin: Technische Universitaet Berlin.

- Beermann, D. y Mihaylov, P. (2014). TypeCraft: Collaborative databasing and resource sharing for linguists. *Language Resources and Evaluation*, Volume 48, Issue 2, 203-225.
- Berners-Lee, T.J., Cailliau, R. y Groff, J-F. (1992). The world-wide web. *Computer Networks and ISDN Systems*, Volume 25, Issues 4–5, November 1992, 454-459.
- Berners-Lee, T., Hendler, J. y Lassila, O. (2001). The Semantic Web. *Scientific American*, May 2001, 29-37. Recuperado el 27 de octubre de 2013 de:
<http://www.scientificamerican.com/article.cfm?id=the-semantic-web>
- Berners-Lee, T. y O’Hara, K. (2013). The read–write Linked DataWeb. *Phil Trans R Soc A*, vol. 371, no. 1987, 1-5.
- Bhole, A., Fortuna, B., Grobelnik, M. y Mladenic', D. (2007). Extracting named entities and relating them over time based on Wikipedia. *Informatica 31: 4 (2007)*, 463-468.
- Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. y Hellmann, S. (2009). DBpedia – A Crystallization Point for the Web of Data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, Issue 7, 154–165.
- Boyd, D. y Crawford, K. (2012). Critical Questions for Big Data: Provocations for a Cultural, Technological, and Scholarly Phenomenon. *Information, Communication, & Society* Vol. 15, No.5, 662-679.
- Brdar, M. y Brdar-Szabó, R. (2007). When Zidane is not simply Zidane, and Bill Gates is not just Bill Gates. Some thoughts on the construction of metaphonymic meanings of proper names. En G. Radden., K.M. Köpcke, T. Berg y P. Siemund (Eds.), *Aspects of Meaning Construction* (pp.125–142). Amsterdam/Philadelphia: John Benjamins.
- Bühler, K. (1985). *Teoría del lenguaje*. Traducción de Julián Marías. 2ª edición. Madrid: Alianza Universidad.

- Butler, C.S. (2009). The Lexical Constructional Model: Genesis, strengths and challenges. En C. Butler y J. M. Arista (Eds.), *Deconstructing Constructions* (pp.117-151). Amsterdam: John Benjamins.
- Butler, C.S. (2012). An ontological approach to the representational lexicon in Functional Discourse Grammar. *Language Sciences* 34, 619-634.
- Carrión Delgado, M. d. G. (2012). Extracción y análisis de unidades léxico-conceptuales del dominio jurídico: un acercamiento metodológico desde FunGramKB. *RaeL* 11, 25-39.
- Carrión Varela, M. d. I. L. (2010). *Representación formal del conocimiento enciclopédico en FunGramKB: el concepto PLACE*. (Trabajo de Fin de Máster). Universidad Nacional de Educación a Distancia, Madrid.
- Cimiano, P., Buitelaar, P., McCrae, J. y Sintek, M. (2011). LexInfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 9 (1), 29-51.
- Corcho, O., Fernández López, M. y Gómez Pérez, A. (2001). *Technical Roadmap v. 1.0, IST-OntoWeb Project*. Madrid: Universidad Politécnica de Madrid.
- Davies, M. (2004-). BYU-BNC. (Basado en el *British National Corpus* de Oxford University Press). Recuperado el 15 de octubre de 2013 de: <http://corpus.byu.edu/bnc>
- Davies, M. (2008-). *The Corpus of Contemporary American English: 450 million words, 1990-present*. Recuperado el 15 de octubre de 2013 de: <http://corpus.byu.edu/coca>
- De Melo, G., Suchanek, F.M. y Pease, A. (2008). Integrating YAGO into the Suggested Upper Merged Ontology. *20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008)*. Dayton, Ohio, USA.
- Ding, L., Finin, T., Joshi, A., Peng, Y., R.S., Sachs, J., Pan, R., Reddivari, P. and Doshi, V. (2004). Swoogle: A Search and Metadata Engine for the Semantic Web. En *Proceedings of the Thirteenth ACM Conference on Information and Knowledge*

- Management. New York (USA), November 2004, (pp. 652-659). New York: ACM.
- Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y. and Kolari, P. (2005). Finding and Ranking Knowledge on the Semantic Web. En *Proceedings of the 4th International Semantic Web Conference*. Galway (Ireland), November 2005 (pp.156-170). Berlin Heidelberg: Springer-Verlag.
- Eckert, M.y Strube, M. (1999). Resolving Discourse Deitic Anaphora in Dialogues. En *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, (pp. 37–44), Bergen, Norway.
- Fan, J., Barker, K. y Porter, B. (2005). Indirect anaphora resolution as semantic path search. En *Proceedings of the 3rd international conference on Knowledge capture (K-CAP '05)*. ACM, New York, NY, USA, (pp.153-160). Recuperado el 28 de noviembre de 2013 de: <http://doi.acm.org/10.1145/1088622.1088650>
- Fass, D. (1991). met*: A method for discriminating metonymy and metaphor by computer. *Computational Linguistics*, Volume 17, Number 1, 49–90.
- Fazzinga, B., Gianforme, G., Gottlob, G. y Lukasiewicz, T. (2011). Semantic Web Search Based on Ontological Conjunctive Queries. *Web Semantics: Science, Services and Agents on the World Wide Web*, Volume 9, Issue 4, December 2011, 453–473.
- Felices Lago, Á. y Ureña Gómez-Moreno, P. (2012). Fundamentos metodológicos de la creación subontológica en FunGramKB. *Onomázein* 26, 49-67.
- Fellbaum, C. (1998, ed.). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.
- Fernández, M.A., Ruiz, J.M., Arráez Jarque, O. y Carrión Varela, M. (2011). Knowledge Living on the Web (KLW). En J. M. Ferrández, J.R. Álvarez Sánchez, F. de la Paz y F.J. Toledo (Eds.), *Lecture Notes in Computer Science 6686. 4th International Work-conference on the Interplay Between Natural and Artificial Computation, IWINAC 2011, La Palma, Canary Islands, Spain, May*

30 - June 3, 2011. *Proceedings, Part II* (pp.391-400). Berlin Heidelberg: Springer-Verlag.

Fernández Fernández, J. y Sacramento Lechado, E. (2012). FunGramKB and its potential applicability to sentiment analysis. Presentación efectuada en *I International Conference on Meaning and Knowledge Representation: LCM & FunGramKB*, Universidad Nacional de Educación a Distancia, Madrid, 4-6 julio 2012.

Frege, G. (1948). Sense and Reference. *The Philosophical Review*, Volume 57, Issue 3 May 1948, 209-230.

Galicia-Haro, S. N., Gelbukh, A. F. y Bolshakov, I. A. (2001). Una aproximación para resolución de ambigüedad estructural empleando tres mecanismos diferentes. *Procesamiento del lenguaje natural*, 27, 55-64.

Giles, J. (2008). Birth pangs for the 'semantic web'. *New Scientist*, Issue 2658, 31 mayo, 26-27.

Goossens, L. (1990). Metaphtonymy: the interaction of metaphor and metonymy in expressions for linguistic action. *Cognitive Linguistics*, 1 (3), 323-340.

Gruber, T. R. (1992). *Ontolingua: A mechanism to support portable ontologies*. Stanford University, Knowledge Systems Laboratory.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, Vol. 5 (2), 199-220.

Guarino, N. y Giaretta, P. (1995). Ontologies and knowledge bases. Towards a terminological clarification. En N.J.I. Mars (Ed.), *Towards Very Large Knowledge Bases* (pp. 25-32). Amsterdam / Tokio: IOS Press.

Halliday, M.A.K. y Hasan, R. (1976). *Cohesion in English*. London: Longman.

Heflin, J. (2004). Web Ontology Language (OWL) Use Cases and Requirements. *World Wide Web Consortium, Recommendation REC-webont-req-20040210*. Recuperado el 18 de abril de 2014 de: <http://www.w3.org/TR/2004/REC-webont-req-20040210/>

- Hendler, J. y Berners-Lee, T. (2010) From the Semantic Web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence*, Volume 174, Issue 2, February 2010,156-161.
- Hernández Terrés, J.M. (1984). *La elipsis en la teoría gramatical*. Murcia: Universidad de Murcia.
- Hoffart, J., Suchanek, F., Berberich, K. y Weikum, G. (2010). *YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia*. Research Report MPI-I-2010-5-007, Max-Planck-Institut für Informatik.
- Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A. y Decker, S. (2011). Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, Volume 9, Issue 4, December 2011, 365-401.
- Hume, D. (1910). An Enquiry Concerning Human Understanding. En *English Philosophers of the Seventeenth and Eighteenth Centuries: Locke, Berkeley, Hume*. Harvard Classics Volume 37. Nueva York: P.F. Collier & Son.
- Jentzsch, A. (2009). DBpedia – Extracting structured data from Wikipedia. Presentación en *Semantic Web In Bibliotheken (SWIB2009)*, Colonia, Alemania, Noviembre 2009. Recuperado el 15 de octubre de 2013 de: http://www.anjajentzsch.de/slides/SWIB09_DBpedia.pdf
- Kasneci, G., Ramanath, M., Suchanek, F. y Weikum, G. (2008). The YAGO-NAGA approach to knowledge discovery. *SIGMOD Record* 37 (4), 41- 47.
- Kifer, M. y Lausen, G. (1989). F-logic: a higher-order language for reasoning about objects, inheritance, and scheme. En J. Clifford, B. Lindsay y D. Maier (Eds.). *Proceedings of the 1989 ACM SIGMOD international conference on Management of data (SIGMOD '89)*, (pp.134-146). New York: ACM.
- King, J.C. (2013). Anaphora. *The Stanford Encyclopedia of Philosophy* (Summer 2013 Edition), Edward N. Zalta (Ed.). Recuperado el 23 de octubre de 2013 de: <http://plato.stanford.edu/archives/sum2013/entries/anaphora/>

- Lakoff, G. (1987). *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*. Chicago / Londres: The University of Chicago Press.
- Lakoff, G. (1993). The contemporary theory of metaphor. En *Metaphor and Thought 2*, (pp.202-251). Cambridge: Cambridge University Press.
- Langacker, R. W. (1987). *Foundations of Cognitive Grammar. Vol. I: Theoretical Prerequisites*. Londres: Longman.
- Langacker, R. (1991). *Foundations of Cognitive Grammar. Vol. 2: Descriptive Application*. Stanford: Stanford University Press.
- Lehmann, J., Furche, T., Grasso, G., Ngonga Ngomo, A.C., Schallhart, C., Sellers, A., Unger, C., Bühmann, L., Gerber, D., Höffner, K., Liu, D. y Auer, S. (2012). DEQA: Deep Web Extraction for Question Answering. En *The Semantic Web – ISWC 2012. 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II* (pp 131-147). Berlin Heidelberg: Springer.
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., y Bizer, C. (en prensa). DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Submitted to the *Semantic Web Journal*, 2013. Recuperado el 18 de abril de 2013 de: http://svn.aksw.org/papers/2013/SWJ_DBpedia/public.pdf
- Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D. y Shepherd, M. (1990). Cyc: toward programs with common sense. *Communications of the ACM*, vol. 33, nº 8, 30-49.
- Liu, H. y Singh, P. (2004a). Commonsense reasoning in and over natural language. En *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2004)*, (pp. 293-306). Wellington, New Zealand: Springer.
- Liu, H. y Singh, P. (2004b). ConceptNet- a practical commonsense reasoning tool-kit. *BT Technology Journal*, vol. 22, nº 4, 211-226.

Longman Dictionary of Contemporary English Online. Recuperado el 6 de diciembre de 2013 de: <http://www.ldoceonline.com>

Mairal Usón, R. (2012). La arquitectura de una base de conocimiento léxico conceptual: implicaciones lingüísticas. En M. Giammatteo, L. Ferrari y H. Albano (Eds.). *Léxico y Sintaxis*. Volumen temático de la serie editada por la Sociedad Argentina de Lingüística. Editorial FFyL, UNCuyo y SAL: Mendoza. ISBN 978-950-774-220-0, (pp. 183-210). Recuperado el 13 de noviembre de 2013 de: <http://ffyl.uncu.edu.ar/spip.php?article3638>.

Mairal Usón, R. y Perriñán Pascual, J.C. (2009). The anatomy of the lexicon within the framework of an NLP knowledge base. *Revista española de lingüística aplicada*, Vol. 22, 217-244.

Mairal Usón, R. y Perriñán Pascual, J.C. (2010a). Role and Reference Grammar and Ontological Engineering. En J.L. Cifuentes, A. Gómez, A. Lillo, J. Mateo y F. Yus (Eds.) *Los caminos de la lengua. Estudios en homenaje a Enrique Alcaraz Varó* (pp. 649-665). Alicante: Universidad de Alicante.

Mairal Usón, R. y Perriñán Pascual, J.C. (2010b). Teoría lingüística y representación del conocimiento: una discusión preliminar. En D. García Padrón y M. d. C. Fumero Pérez (Eds.), *Tendencias en lingüística general y aplicada* (pp. 155-168). Berlín: Peter Lang.

Mairal Usón, R., Perriñán Pascual, C. y Pérez Cabello de Alba, M. B. (2012). La representación léxica. Hacia un enfoque ontológico. En R. Mairal Usón, L. Guerrero y C. González (Eds.) *El funcionalismo en la teoría lingüística. La Gramática del Papel y la Referencia. Introducción, avances y aplicaciones* (pp. 85-102). Madrid: Akal.

Mairal Usón, R. y Ruiz de Mendoza Ibáñez, F. (2009). Levels of description and explanation in meaning construction. En C. Butler y J. Martín Arista (Eds.), *Deconstructing Constructions* (pp. 153-198). Amsterdam/Philadelphia: John Benjamins.

Martin, J.R. (1992). *English Text: System and Structure*. Philadelphia/Amsterdam: John Benjamins.

- Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, Vol. 38, No. 11, 39-41.
- Minsky, M. (1974). A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306, June, 1974*.
- Minsky, M. (2000). Commonsense-based interfaces. *Communications of the ACM* 43, nº 8, 67-73.
- Mitkov, R. (2002). *Anaphora Resolution*. Pearson Education Limited. Great Britain: Longman.
- Morse, M., Lehmann, J., Auer, S., Stadler, C. y Hellmann, S. (2012). DBpedia and the live extraction of structured data from Wikipedia. *Program: electronic library and information systems*, Vol. 46, Issue 2, 157 – 181.
- Noy, N. F. y McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report KSL-01-05*. Stanford Knowledge Systems Laboratory. Stanford: Stanford University.
- Palomar Sanz, M., Saiz Noeda, M., Muñoz Guillena, R., Suárez Cueto, A. y Martínez Barco, P. (2000). PHORA: a system to solve the anaphora in Spanish. En *Proceedings of Third International Conference on Discourse Anaphora and Anaphor Resolution (DAARC2000)*, (pp. 206-211). Lancaster: University of Lancaster.
- Pang, K.S. (2010). Eponymy and life-narratives: The effect of foregrounding on proper names. *Journal of Pragmatics*, Volume 42, Issue 5, May 2010, 1321-1349.
- Panton, K., Matuszek, C., Lenat, D., Schneider, D., Witbrock, M., Siegel, N. y Shepard, B. (2006). Common sense reasoning - from Cyc to intelligent assistant. En Yang Cai y Julio Abascal (Eds.), *Ambient Intelligence in Everyday Life* (pp. 1-31). Berlin: Springer.
- Paredes Duarte, M.J. (2008). El principio de economía lingüística. *Pragmalingüística* 15-16 (2007-08) 7-27, 166-178.

- Peña Cervel, M. S. (2011). Macbeth Revisited: A Cognitive Analysis. *Metaphor and Symbol*, 26: 1, 1-22.
- Periñán Pascual, C. (2012a). En defensa del procesamiento del lenguaje natural fundamentado en la lingüística teórica. *Onomázein*, 2 (26), 13-48.
- Periñán Pascual, C. (2012b). The situated common-sense knowledge in FunGramKB. *Review of Cognitive Linguistics* 10 (1), 184-214.
- Periñán Pascual, C. y Arcas Túnez, F. (2004). Meaning postulates in a lexico-conceptual knowledge base. *15th International Workshop on Databases and Expert Systems Applications, IEE, Los Alamitos (California)*, 38-42.
- Periñán Pascual, J. C. y Arcas Túnez, F. (2005). Microconceptual-Knowledge Spreading in FunGramKB. En *Proceedings on the Ninth IASTED International Conference on Artificial Intelligence and Soft Computing*, (pp. 239-244). Anaheim-Calgary-Zurich: ACTA Press.
- Periñán Pascual, J. C. y Arcas Túnez, F. (2006). Reusing Computer-oriented Lexica as Foreign-Language Electronic Dictionaries. *Anglogermánica online: Revista electrónica periódica de filología alemana e inglesa*, nº 4, 69-93.
- Periñán Pascual, C. y Arcas Túnez, F. (2007a). Cognitive modules of an NLP knowledge base for language understanding. *Procesamiento del Lenguaje Natural*, nº 39, 197-204.
- Periñán Pascual, C. y Arcas Túnez, F. (2007b). Deep semantics in an NLP knowledge base. En D. Borrajo, L. Castillo y J. M. Corchado (Eds.), *12th Conference of the Spanish Association for Artificial Intelligence* (pp. 279-288). Salamanca: Universidad de Salamanca.
- Periñán Pascual, C. y Arcas Túnez, F. (2010a). Ontological commitments in FunGramKB. *Procesamiento del Lenguaje Natural*, nº 44, 27-34.
- Periñán Pascual, C. y Arcas Túnez, F. (2010b). The Architecture of FunGramKB. En N. Calzolari *et al.* (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, (pp. 2667- 2674).Valletta, Malta: European Language Resources Association (ELRA).

- Periñán Pascual, C. y Arcas Túnez, F. (2014). La ingeniería del conocimiento en el dominio legal: La construcción de una Ontología Satélite en FunGramKB. *Revista Signos: Estudios de Lingüística* 47 (84), 113-139.
- Periñán Pascual, C. y Carrión Varela, M.d.l.L. (2011). FunGramKB y el conocimiento cultural. En *Anglogermánica Online 2011*, 87-105.
- Periñán Pascual, C. y Mairal Usón, R. (2009). Bringing Role and Reference Grammar to natural language understanding. *Procesamiento del lenguaje natural*, nº 43, 265-273.
- Periñán Pascual, C. y Mairal Usón, R. (2010a). Enhancing UniArab with FunGramKB. *Procesamiento del Lenguaje Natural*, Revista nº 44, marzo de 2010, 19-26.
- Periñán Pascual, C. y Mairal Usón, R. (2010b). La gramática de COREL: un lenguaje de representación conceptual. *Onomázein* 21.
- Periñán-Pascual, C y Mairal Usón, R. (2011). The COHERENT Methodology in FunGramKB. *Onomázein: Revista de lingüística, filología y traducción de la Pontificia Universidad Católica de Chile*, nº. 24, 13-33.
- Periñán Pascual, C. y Mairal Usón, R. (2012). La dimensión computacional de la Gramática del Papel y la Referencia: la estructura lógica conceptual y su aplicación en el procesamiento del lenguaje natural. En R. Mairal Usón, L. Guerrero y C. González (eds.) *El funcionalismo en la teoría lingüística. La Gramática del Papel y la Referencia. Introducción, avances y aplicaciones* (pp. 333-348). Madrid: Akal.
- Procter, P. (ed.) (1978). *Longman Dictionary of Contemporary English*. Harlow (Essex): Longman.
- REAL ACADEMIA ESPAÑOLA. Banco de datos (CREA) [en línea]. *Corpus de referencia del español actual*. Recuperado el 21 de octubre de 2013 de: <http://www.rae.es>
- REAL ACADEMIA ESPAÑOLA. (2001). *Diccionario de la lengua española (DRAE)*. 22.ª Edición. Versión electrónica, recuperada el 6 de noviembre de 2013 de: <http://www.rae.es/recursos/diccionarios/drae>

- REAL ACADEMIA ESPAÑOLA. (2005). *Diccionario panhispánico de dudas (DPD)*, 1.ª edición. Versión electrónica, recuperada el 10 de mayo de 2014 de: <http://www.rae.es/recursos/diccionarios/dpd>
- Ruiz de Mendoza Ibáñez, F. J. (1999). *Introducción a la teoría cognitiva de la metonimia [An introduction to the cognitive theory of metonymy]*. Granada: Granada Lingüística y Método Ediciones.
- Ruiz de Mendoza Ibáñez, F.J. (2000). The role of mappings and domains in understanding metonymy. En A. Barcelona (Ed.) *Metaphor and Metonymy at the Crossroads*. (pp. 109-132). Berlin/New York: Mouton de Gruyter.
- Ruiz de Mendoza Ibáñez, F.J. y Galera-Masegosa, A. (2011). Going beyond metaphonymy: Metaphoric and metonymic complexes in phrasal verb interpretation. *Language Value*, December 2011, Volume 3, Number 1, 1-29.
- Ruiz de Mendoza Ibáñez, F.J. y Mairal Usón, R. (2008). Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model. *Folia Linguistica*, Volume 42, Issue 3-4, 355–400.
- Rosch, E.H. (1973). Natural categories. *Cognitive Psychology* 4, 328-50.
- Saussure, F. D. (1945). *Curso de Lingüística General*. Traducción de Amado Alonso. 24ª edición. Buenos Aires: Editorial Losada.
- Shutova, E. (2010). Models of metaphor in NLP. En *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 688-697), Uppsala, Sweden: Association for Computational Linguistics.
- Stojanovic, L., Stojanovic, N. y Volz, R. (2002). Migrating data-intensive web sites into the Semantic Web. En *Proceedings of the 2002 ACM symposium on Applied computing (SAC '02)* NY, USA, (pp. 1100-1107). New York: ACM.
- Strohmaier, M., Walka, S., Pöschko, J., Lamprechta, D., Tudorache, T., Nyulas, C., Musenb, M.A. y Noy, N. F. (2013). How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 20 (2013), 18-34.

- Stumme, G., Hotho, A. and Berendt, B. (2006). Semantic Web Mining: State of the art and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*. Volume 4, Issue 2, June 2006, 124-143.
- Suchanek, F., Kasneci, G. y Weikum, G. (2007). YAGO: A Core of Semantic Knowledge - Unifying WordNet and Wikipedia. *16th International World Wide Web Conference (WWW 2007)*, 697-706.
- Suchanek, F., Kasneci, G. y Weikum, G. (2008). YAGO: A Large Ontology from Wikipedia and WordNetWeb. *Semantics: Science, Services and Agents on the World Wide Web*. Volume 6, Issue 3, September 2008 (pp. 203-217), World Wide Web Conference 2007 Semantic Web Track.
- Thalheim, B. y Wang, Q. (2013). Data migration: A theoretical perspective. *Data & Knowledge Engineering*, Volume 87, September 2013, 260-278.
- Unger, C. y Cimiano, P. (2011). Pythia: Compositional Meaning Construction for Ontology-Based Question Answering on the Semantic Web. En R. Muñoz A. Montoyo y E. Metais (Eds.), *Natural Language Processing and Information Systems. 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011, Alicante, Spain, June 28-30, 2011, Proceedings* (pp.153-160). Berlin - Heidelberg: Springer.
- Ureña Gómez-Moreno, P., Alameda Hernández, Á. y Felices Lago, Á. (2011). Towards a specialised corpus of organized crime and terrorism. En M.L. Carrió *et al.* (Eds.) *La investigación y la enseñanza aplicadas a las lenguas de especialidad y a la tecnología* (pp. 301-306). Valencia: Universitat Politècnica de Valencia.
- Uschold, M. y Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11, nº 2, 93-136.
- Van Valin, R.D. Jr. y Mairal Usón, R. (en prensa). Interfacing the Lexicon and an Ontology in a Linking Algorithm. En M. Ángeles Gómez, F. Ruiz de Mendoza y F. González-García (Eds.), *Form and Function in Language: Functional, Cognitive and Applied Perspectives. Essays in Honour of Christopher S. Butler*. Amsterdam: John Benjamins.

- Vázquez, S. (2009). *Resolución de la ambigüedad semántica mediante métodos basados en conocimiento y su aportación a tareas de PLN*. (Tesis Doctoral). Depto. de Lenguajes y Sistemas Informáticos. Universidad de Alicante. Alicante, 270.
- Wilks, Y. (1978). Making preferences more active. *Artificial Intelligence*, Volume 11, Issue 3, December 1978, 197-223.
- Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, Volume 8, Issue 3, June 1965, 338-353.
- Zadeh, L.A. (1968). Fuzzy algorithms. *Information and Control*, Volume 12, Issue 2, February 1968, 94-102.

APÉNDICE I

Listado de reglas y propiedades en el Onomasticón de FunGramKB

Domain	Superclass	Property	COREL	Type
Bridge	Building	locatedInArea	+(e1:+BE_00 (x1: <C>)Theme (x2: +BRIDGE_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location)	0
Bridge	Building	mainspan	+(e1:+BE_00 (x1: <C>)Theme (x2:+BRIDGE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+LONG_00)Attribute (f1:<VAL> +METRE_00)Quantity)	0
Bridge	Building	UnitedStates National BridgeID	n/r	
Building	Place	address	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+ADDRESS_00)Attribute (f1: <VAL>)Location)	0
Building	Place	architect	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:\$DESIGN_00 (x3: <VAL>)Theme (x4)Referent (f1: (e3: +BUILD_00 (x3)Theme (x2: +BUILDING_00)Referent))Purpose)	0
Building	Place	architectual Bureau	n/r	
Building	Place	architectural Style	+(e1:+BE_00(x1: <C>)Theme(x2:+BUILDING_00)Referent) +(e2:+BE_01(x1)Theme(x3:<VAL> +STYLE_00)Attribute)	0
Building	Place	awards	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+BE_01 (x1)Theme (x3: +PRIZE_00)Attribute (f1: <VAL>)Quantity)	0
Building	Place	buildingEndDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+BUILD_00 (x3)Theme (x1)Referent) +(e3:+FINISH_00(x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1
Building	Place	buildingStartDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+BUILD_00 (x3)Theme (x1)Referent) +(e3:+BEGIN_00 (x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1

Building	Place	closingDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+CLOSE_00 (x3)Agent (x1)Theme (x4)Location(x5)Origin(x6)Goal(f1: <VAL> +DATE_00)Time)	1
Building	Place	cost	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+COST_00 (x1)Theme (x3:<VAL> \$DOLLAR_00)Attribute)	0
Building	Place	demolitionDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:\$DEMOLISH_00 (x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1
Building	Place	destructionDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+DESTROY_00 (x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1
Building	Place	floorArea	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+BE_02 (x3:+FLOOR_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$\$SQUARE_METRE_00)Quantity)	0
Building	Place	floorCount	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+BE_02 (x3:+FLOOR_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x1)Theme (x3: +FLOOR_00)Attribute (f1: <VAL>)Quantity)	0
Building	Place	formerName	a/n	
Building	Place	openingDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:ing +BE_01 (x1)Theme (x3: +OPEN_00)Attribute (f1: <VAL> +DATE_00)Time)	1
Building	Place	owner	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+HAVE_00 (x3: <VAL>)Theme (x1)Referent)	0
Building	Place	owning Organisation	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+HAVE_00 (x3: <VAL>)Theme (x1)Referent) +(e3: +BE_01 (x3)Theme (x4: +ORGANIZATION_00)Attribute)	0

Building	Place	rebuildingDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) +(e2:+DAMAGE_00 (x3)Theme (x1)Referent) +(e3:+BUILD_00 (x4)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1
Building	Place	structuralSystem	n/r	
Building	Place	tenant	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+LIVE_01 (x3: <VAL>)Theme (x1)Location)	0
Building	Place	visitorsPerYear	n/r	
Building	Place	visitorsTotal	+(e1: +BE_00 (x1: <C>)Theme (x2: +BUILDING_00)Referent) *(e2: past +VISIT_00 (x3)Agent (x4: <VAL> +HUMAN_00)Theme (x5)Location (x6)Origin (x1)Goal)	0
Cave	Place	numberOfEntrances	n/r	
City	PopulatedPlace	administrativeCollectivity	n/r	
City	PopulatedPlace	administrativeDistrict	n/r	
City	PopulatedPlace	annualTemperature	n/r	
City	PopulatedPlace	associationOfLocalGovernment	n/r	
City	PopulatedPlace	coastLine	+(e1:+BE_00 (x1: <C>)Theme (x2:+CITY_00)Referent) +(e2:+BE_02 (x3:+COAST_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+LENGTH_00)Attribute (f1:<VAL> +METRE_00)Quantity)	0
City	PopulatedPlace	cornishName	a/n	
City	PopulatedPlace	country	+(e1:+BE_00 (x1: <C>)Theme (x2: +CITY_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location) +(e3:+BE_00 (x3)Theme (x4: +COUNTRY_00)Referent)	0
City	PopulatedPlace	crownDependency	n/r	

City	PopulatedPlace	day	+(e1:+BE_00 (x1: <C>)Theme (x2:+CITY_00)Referent) *(e2:+SHOW_00 (x3)Theme (x4: (e3: +RESPECT_00 (x5)Theme(x2)Referent))Referent(f1: <VAL> +DATE_00)Time)	0
City	PopulatedPlace	department	n/r	
City	PopulatedPlace	destructionDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+CITY_00)Referent) +(e2:+DESTROY_00 (x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1
City	PopulatedPlace	distanceTo Belfast	n/r	
City	PopulatedPlace	distanceTo Cardiff	n/r	
City	PopulatedPlace	distanceTo Douglas	n/r	
City	PopulatedPlace	distanceTo Edinburgh	n/r	
City	PopulatedPlace	distanceTo London	n/r	
City	PopulatedPlace	federalState	n/r	
City	PopulatedPlace	frazioni	n/r	
City	PopulatedPlace	gaelicName	a/n	
City	PopulatedPlace	irishName	a/n	
City	PopulatedPlace	jointCommunity	n/r	
City	PopulatedPlace	largestMetro	n/r	
City	PopulatedPlace	manxName	a/n	
City	PopulatedPlace	province	n/r	
City	PopulatedPlace	region	n/r	
City	PopulatedPlace	region1	n/r	
City	PopulatedPlace	saint	+(e1:+BE_00 (x1: <C>)Theme (x2:+CITY_00)Referent) *(e2:+PROTECT_00 (x3: <VAL> +SAINT_00)Theme (x1)Referent)	0
City	PopulatedPlace	scottishName	a/n	

City	PopulatedPlace	summer Temperature	n/r	
City	PopulatedPlace	twinCity	n/r	
City	PopulatedPlace	twinCountry	n/r	
City	PopulatedPlace	welshName	a/n	
City	PopulatedPlace	winter Temperature	n/r	
Country	PopulatedPlace	anthem	+(e1:+BE_00 (x1: <C>)Theme (x2: +COUNTRY_00)Referent) *(e2: +BE_00 (x3: \$ANTHEM_00)Theme (x4:<VAL>)Referent) *(e3: +PLAY_01 (x5)Theme (x4)Referent (f1: (e4: +RESPECT_00 (x5)Theme (x1)Referent)))Purpose)	0
Country	PopulatedPlace	currency	+(e1:+BE_00 (x1: <C>)Theme (x2: +COUNTRY_00)Referent) *(e2:+USE_00 (x3)Theme (x4: <VAL>)Referent(f1: x1)Location) +(e3:+BE_00 (x4)Theme (x5: +MONEY_00)Referent)	0
Country	PopulatedPlace	ethnicGroupsIn Year	n/r	
Country	PopulatedPlace	governmentType	+(e1:+BE_00 (x1: <C>)Theme (x2: +COUNTRY_00)Referent) *(e2:+BE_00 (x3)Theme (x4: +GOVERNMENT_00)Referent (f1:x1)Location) *(e3: +BE_00 (x4)Theme (x5: <VAL>)Referent)	0
HistoricPlace	Place	added	n/r	
HistoricPlace	Place	architect	n/r	
HistoricPlace	Place	numberOf Visitors	n/r	
Hospital	Building	bedCount	+(e1:+BE_00 (x1: <C>)Theme (x2:+HOSPITAL_00)Referent) *(e2:+BE_02 (x3: <VAL> +BED_00)Theme (x1)Location (f1:+IN_00)Position)	0
Hospital	Building	country	+(e1:+BE_00 (x1: <C>)Theme (x2:+HOSPITAL_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location) +(e3:+BE_00 (x3)Theme (x4:+COUNTRY_00)Referent)	0
Hospital	Building	region	+(e1:+BE_00 (x1: <C>)Theme (x2:+HOSPITAL_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location)	0

Hospital	Building	state	n/r	
Island	Place	archipelago	+(e1:+BE_00 (x1: <C>)Theme (x2:+ISLAND_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location) +(e3:+BE_00 (x3)Theme(x4:\$ARCHIPELAGO_00)Referent)	0
Island	Place	country	+(e1:+BE_00 (x1: <C>)Theme (x2:+ISLAND_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location) +(e3:+BE_00 (x3)Theme (x4:+COUNTRY_00)Referent)	0
Island	Place	majorIsland	n/r	
Island	Place	numberOf Islands	n/r	
LaunchPad	Building	associatedRocket	n/r	
LaunchPad	Building	firstLaunchDate	n/r	
LaunchPad	Building	firstLaunch Rocket	n/r	
LaunchPad	Building	lastLaunchDate	n/r	
LaunchPad	Building	lastLaunchRocket	n/r	
LaunchPad	Building	launches	n/r	
LaunchPad	Building	launchSite	n/r	
LaunchPad	Building	maximum Inclination	n/r	
LaunchPad	Building	minimum Inclination	n/r	
LaunchPad	Building	numberOfPads	n/r	
LaunchPad	Building	operator	n/r	
LaunchPad	Building	status	n/r	
LaunchPad	Building	totalLaunches	n/r	
LunarCrater	Place	diameter	n/r	
Municipality	PopulatedPlace	neighboring Municipality	+(e1:+BE_00 (x1: <C>)Theme (x2:+TOWN_00)Referent) *(e2:+BE_02 (x3: <VAL>)Theme (x1)Location (f1:+NEAR_00)Position)	0

Organisation	Thing	affiliation	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+CONTROL_00 (x3:<VAL>)Theme (x1)Referent) *(e3:+HAVE_00 (x3)Theme (x1)Referent)	0
Organisation	Thing	endowment	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2: +GIVE_00 (x3)Agent (x2:<VAL> \$DOLLAR_00)Theme (x3)Origin (x1)Goal (f1:(e3: +WORK_00 (x1)Theme))Purpose)	0
Organisation	Thing	foundationDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) +(e2:+CREATE_00 (x3)Theme(x1)Referent(f1:<VAL> +DATE_00)Time)	1
Organisation	Thing	foundation Organisation	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) +(e2:+CREATE_00 (x3: <VAL>)Theme (x1)Referent) +(e3:+BE_01 (x3)Theme (x4: +ORGANIZATION_00)Attribute)	1
Organisation	Thing	foundationPerson	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) +(e2:+CREATE_00 (x3: <VAL>)Theme (x1)Referent) +(e3:+BE_01 (x3)Theme (x4: +HUMAN_00)Attribute)	1
Organisation	Thing	foundationPlace	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) +(e2:+CREATE_00 (x3)Theme(x1)Referent(f1:<VAL>)Location)	1
Organisation	Thing	genre	n/r	
Organisation	Thing	keyPerson	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+WORK_01 (x3: <VAL>)Theme(f1: x1)Location) *(e3:+BE_01 (x3)Theme (x4:m +IMPORTANT_00)Attribute)	0
Organisation	Thing	keyPerson position	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+WORK_01 (x3)Theme (f1: x1)Location) *(e3:+BE_01 (x3)Theme (x4:m +IMPORTANT_00)Attribute) *(e4: +DO_00 (x3)Theme (x5:<VAL>)Referent (f2: (e5:+BE_00(x5)Theme (x6:+JOB_00)Referent))Condition)	0
Organisation	Thing	location	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location)	0

Organisation	Thing	locationCity	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +CITY_00)Attribute)	0
Organisation	Thing	locationCountry	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_02 (x1)Theme (x3: <VAL>)Location) +(e3:+BE_01 (x3)Theme (x4: +COUNTRY_00)Attribute)	0
Organisation	Thing	motto	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+BE_01 (x3:+SENTENCE_00)Theme (x4: <VAL>)Attribute(f1:x1)Referent)	0
Organisation	Thing	numberOf Employees	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+WORK_01 (x3)Theme (f1:<VAL>)Quantity(f2: x1)Location)	0
Organisation	Thing	owningCompany	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+HAVE_00 (x3:<VAL>)Theme (x1)Referent) +(e3:+BE_01 (x3)Theme (x4: +COMPANY_00)Attribute)	0
Organisation	Thing	postalCode	n/r	
Organisation	Thing	product	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+SELL_00 (x1)Agent (x3: <VAL>)Theme (x4)Origin (x5)Goal)	0
Organisation	Thing	revenue	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+OBTAIN_00 (x1)Theme (x3: <VAL> \$DOLLAR_00)Referent)	0
Organisation	Thing	slogan	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2:+USE_00 (x1)Theme (x3: <VAL>)Referent(f1: (e3:+SELL_00 (x1)Agent (x4: m +PRODUCT_00)Theme (x5)Origin (x6)Goal))Purpose) +(e4:+BE_00 (x3)Theme (x7:+SENTENCE_00)Referent)	0
Organisation	Thing	type	+(e1:+BE_00 (x1: <C>)Theme (x2:+ORGANIZATION_00)Referent) *(e2: +BE_01 (x1)Theme (x3:<VAL>)Attribute)	0
Person	Thing	activeYearsEnd Date	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: egr +WORK_00 (x1)Theme (f1:<VAL> +DATE_00)Time)	1

Person	Thing	activeYearsStart Date	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: ing +WORK_00 (x1)Theme (f1:<VAL> +DATE_00)Time)	1
Person	Thing	almaMater	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: past +EXAMINE_00 (x1)Theme (x3)Referent(f1:<VAL> +UNIVERSITY_00)Location)	0
Person	Thing	award	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: past +WIN_00 (x1)Theme (x3:<VAL>)Referent)	0
Person	Thing	birthDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: ing +LIVE_00(x1)Theme (f1: <VAL> +DATE_00)Time)	1
Person	Thing	birthName	a/n	
Person	Thing	birthPlace	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: ing +LIVE_00(x1)Theme (f1: <VAL>)Location)	1
Person	Thing	boards	n/r	
Person	Thing	child	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +BE_01 (x3: <VAL>)Theme (x4: +CHILD_00)Attribute(f1:x1)Referent)	0
Person	Thing	citizenship	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+LIVE_01 (x1)Theme (x3: <VAL>)Location(f1:+LEGAL_00)Manner(f2:(e3:ing +LIVE_00 (x1)Theme (f3:x3)Location))Reason (f4: (e4: +LIVE_01 (x1)Theme (x3)Location (f5: m +YEAR_00)Duration))Reason) +(e5:+BE_01 (x3)Theme (x4:+COUNTRY_00)Attribute)	0
Person	Thing	deathCause	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +DIE_00 (x1)Theme (f1: <VAL>)Reason)	1
Person	Thing	deathDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +DIE_00 (x1)Theme (f1: <VAL>)Time)	1
Person	Thing	deathPlace	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +DIE_00 (x1)Theme (f1: <VAL>)Location)	1

Person	Thing	education	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +DO_00 (x1)Theme (x3: <VAL> +STUDY_00)Referent)	0
Person	Thing	employer	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +WORK_01 (x1)Theme (f1: (e3: +PAY_00 (x3: <VAL>)Agent (x4)Theme (x3)Origin (x1)Goal))Purpose)	0
Person	Thing	ethnicity	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2:+BE_01 (x1)Theme (x3: <VAL>)Attribute)	0
Person	Thing	father	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +BE_01 (x3: <VAL>)Theme (x4: +FATHER_00)Attribute(f1:x1)Referent)	0
Person	Thing	height	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+HIGH_00)Attribute (f1: <VAL> \$CENTIMETER_00)Quantity)	0
Person	Thing	homeTown	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: ing +LIVE_00 (x1)Theme(f1: <VAL>)Location) *(e3: +LIVE_01 (x1)Theme (x3: <VAL>)Location (f2:+CHILDHOOD_00)Duration)	1
Person	Thing	influenced	n/r	
Person	Thing	influencedBy	n/r	
Person	Thing	knownFor	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +KNOW_00 (x3)Theme (x1)Referent (f1: <VAL>)Reason)	0
Person	Thing	mother	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2: +BE_01 (x3: <VAL>)Theme (x4: +MOTHER_00)Attribute(f1:x1)Referent)	0
Person	Thing	nationality	n/r	
Person	Thing	networth	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+BE_01 (x1)Theme (x3: +VALUE_00)Attribute(f1:<VAL> \$DOLLAR_00)Quantity)	0

Person	Thing	occupation	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+DO_00 (x1)Theme (x3:<VAL>)Referent (f1:(e3:+BE_00 (x3)Theme (x4:+JOB_00)Referent)))Condition)	0
Person	Thing	otherName	a/n	
Person	Thing	partner	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+BE_01 (x3: <VAL>)Theme (x4:+SPOUSE_00)Attribute(f1:x1)Referent)	0
Person	Thing	party	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+WORK_01 (x1)Theme (f1:<VAL> \$PARTY_01)Beneficiary)	0
Person	Thing	predecessor	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+TRANSFER_00 (x3)Agent (x4: +JOB_00 +OCCURRENCE_00 +FAME_00 +MONEY_00)Theme (x5: <VAL>)Origin (x1)Goal)	1
Person	Thing	pseudonym	a/n	
Person	Thing	relation	n/r	
Person	Thing	relative	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2:+BE_01 (x3: <VAL>)Theme (x4:+RELATIVE_00)Attribute(f1:x1)Referent)	0
Person	Thing	religion	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2:+BE_01 (x1)Theme (x3:+RELIGION_00)Attribute) *(e3:+BE_00 (x3)Theme (x4: <VAL>)Referent)	0
Person	Thing	residence	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+LIVE_01 (x1)Theme(x3:<VAL>)Location)	0
Person	Thing	restingPlace	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) +(e2:+DIE_00 (x1)Theme) *(e3: +COVER_00 (x3:+HUMAN_00)Agent (x4: +GROUND_00 +STONE_00)Theme (x5)Origin (x1)Goal (f1:+ON_00)Position (f2:<VAL>)Location)	0
Person	Thing	restingPlace Coordinates	n/r	

Person	Thing	salary	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+REMUNERATE_00 (x3:+HUMAN_00)Agent (x4: <VAL> \$DOLLAR_00)Theme (x5)Origin (x1)Goal)	0
Person	Thing	spouse	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +MARRY_00 (x1)Theme (x3: <VAL>)Referent)	0
Person	Thing	successor	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2: +TRANSFER_00 (x3)Agent (x4: +JOB_00 +OCCURRENCE_00 +FAME_00 +MONEY_00)Theme (x1)Origin (x5: <VAL>)Goal)	1
Person	Thing	term	n/r	
Person	Thing	title	n/r	
Person	Thing	weight	+(e1:+BE_00 (x1: <C>)Theme (x2:+HUMAN_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+WEIGHT_00)Attribute (f1: <VAL> \$KILOGRAM_00)Quantity)	0
Place	Thing	areaLand	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *((e2:+BE_02 (x3:+LAND_00)Theme (x1)Location (f1:+IN_00)Position)(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$\$SQUARE_METRE_00)Quantity))	0
Place	Thing	areaTotal	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+BIG_00)Attribute (f1: <VAL> \$\$SQUARE_METRE_00)Quantity)	0
Place	Thing	areaWater	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *((e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position)(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$\$SQUARE_METRE_00)Quantity))	0
Place	Thing	coordinates	n/r	
Place	Thing	depth	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity)	0

Place	Thing	elevation	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)	0
Place	Thing	height	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+HIGH_00)Attribute (f1: <VAL> +METRE_00)Quantity)	0
Place	Thing	length	+(e1: +BE_00 (x1: <C>)Theme (x2: +PLACE_00)Referent) *(e2: +BE_01 (x1)Theme (x3: +LONG_00)Attribute (f1: <VAL> +METRE_00)Quantity)	0
Place	Thing	location	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location)	0
Place	Thing	maximumDepth	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: +DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m +DEEP_00)Attribute))Scene)	0
Place	Thing	maximum Elevation	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m \$ALTITUDE_00)Attribute))Scene)	0
Place	Thing	minimum elevation	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: p \$ALTITUDE_00)Attribute))Scene)	0
Place	Thing	nativeName	a/n	
Place	Thing	nearestCity	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme (x1)Location (f1: m +NEAR_00)Position)	0
Place	Thing	nickname	a/n	

Place	Thing	otherName	a/n	
Place	Thing	percentageOfAreaWater	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *((e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position)(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$PERCENTAGE_00)Quantity))	0
Place	Thing	type	n/r	
Place	Thing	width	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+WIDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)	0
PopulatedPlace	Place	areaCode	n/r	
PopulatedPlace	Place	areaMagnitude	n/r	
PopulatedPlace	Place	areaMetro	n/r	
PopulatedPlace	Place	areaUrban	n/r	
PopulatedPlace	Place	capital	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3: <VAL>)Theme (x1)Location(f1:(e3:+BE_00(x3)Theme(x4:+CITY_00)Referent))Condition) +(e4:+BE_01 (x3)Theme (x5: \$IMPORTANT_01)Attribute)	0
PopulatedPlace	Place	censusYear	n/r	
PopulatedPlace	Place	demonym	n/r	
PopulatedPlace	Place	establishedTitle	n/r	
PopulatedPlace	Place	ethnicGroup	n/r	
PopulatedPlace	Place	foundingDate	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+CREATE_00 (x3)Theme (x1)Referent (f1:<VAL> +DATE_00)Time)	1
PopulatedPlace	Place	foundingPerson	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+CREATE_00 (x3:<VAL>)Theme (x1)Referent) +(e3:+BE_01 (x3)Theme (x4: +HUMAN_00)Attribute)	0

PopulatedPlace	Place	foundingYear	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+CREATE_00 (x3)Theme (x1)Referent (f1:<VAL> +YEAR_00)Time)	1
PopulatedPlace	Place	language	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2: +LIVE_01 (x3:+POPULATION_00)Theme (x1)Location) *(e3:+SAY_00 (x3)Theme (x4)Referent (x5)Goal (f1: <VAL> +LANGUAGE_00)Instrument)	0
PopulatedPlace	Place	languageType	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2: +LIVE_01 (x3:+POPULATION_00)Theme (x1)Location) *(e3:+SAY_00 (x3)Theme (x4)Referent (x5)Goal(f1: (e4: +BE_01 (x6:+LANGUAGE_00)Theme (x7: <VAL>)Attribute))Instrument)	0
PopulatedPlace	Place	largestCity	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme(x1)Location (f1:+IN_00)Position) +(e3:+BE_01 (x3)Theme (x4: +CITY_00 & m +BIG_00)Attribute)	0
PopulatedPlace	Place	largestSettlement	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme(x1)Location (f1:+IN_00)Position) +(e3:+BE_01 (x3)Theme (x4: +ARTIFICIAL_AREA_00 & m +BIG_00)Attribute)	0
PopulatedPlace	Place	largestSettlement type	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3: +ARTIFICIAL_AREA_00 & m +BIG_00)Theme (x1)Location (f1:+IN_00)Position) +(e3:+BE_01 (x3)Theme (x4:<VAL>)Attribute)	0
PopulatedPlace	Place	leaderName	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+GOVERN_00 (x3: <VAL>)Theme (x1)Referent)	0
PopulatedPlace	Place	leaderParty	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2: +GOVERN_00 (x3)Theme (x1)Referent) *(e3:+WORK_01 (x3)Theme (f1:<VAL> \$PARTY_01)Beneficiary)	0
PopulatedPlace	Place	leaderTitle	n/r	

PopulatedPlace	Place	motto	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x3:+SENTENCE_00)Theme (x4: <VAL>)Attribute (f1:x1)Referent)	0
PopulatedPlace	Place	populationAsOf	n/r	
PopulatedPlace	Place	populationDensity	n/r	
PopulatedPlace	Place	populationMetro	n/r	
PopulatedPlace	Place	populationMetro Density	n/r	
PopulatedPlace	Place	populationTotal	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+LIVE_01 (x3:+POPULATION_00)Theme (x1)Location (f1:<VAL>)Quantity)	0
PopulatedPlace	Place	populationUrban	n/r	
PopulatedPlace	Place	populationUrban Density	n/r	
PopulatedPlace	Place	postalCode	n/r	
PopulatedPlace	Place	regionalLanguage	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x1)Theme (x3: +COUNTRY_00)Location (f1:+IN_00)Position) +(e3:+LIVE_01 (x4:+POPULATION_00)Theme (x1)Location) *(e4:+SAY_00 (x4)Theme (x5)Referent (x6)Goal (f2: <VAL> +LANGUAGE_00)Instrument)	0
ProtectedArea	Place	iucnCategory	n/r	
ProtectedArea	Place	numberOfVisitors	n/r	
Skyscraper	Building	contractor	+(e1:+BE_00 (x1: <C>)Theme (x2:\$SKYSCRAPER_00)Referent) +(e2: +BUILD_00 (x3: <VAL>)Theme (x1)Referent)	0
Skyscraper	Building	developer	n/r	
Skyscraper	Building	engineer	+(e1:+BE_00 (x1: <C>)Theme (x2:\$SKYSCRAPER_00)Referent) +(e2:\$DESIGN_00 (x3: <VAL>)Theme (x1)Referent)	0

Skyscraper	Building	highestBuildingIn Year	n/r	
Skyscraper	Building	management	+(e1:+BE_00 (x1: <C>)Theme (x2:+BUILDING_00)Referent) *(e2:+CONTROL_00 (x3: <VAL>)Theme (x1)Referent)	0
Station	Building	platform	n/r	
WineRegion	Place	growingGrape	n/r	
WineRegion	Place	numberOf Vineyards	n/r	
WineRegion	Place	numberOf Wineries	n/r	
WineRegion	Place	partOfWine Region	n/r	
WineRegion	Place	similar	n/r	
WineRegion	Place	subregion	n/r	
WineRegion	Place	varietals	n/r	
WineRegion	Place	wineProduced	n/r	
WineRegion	Place	wineYear	n/r	
Work	Thing	author	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent +(e2: +CREATE_00 (x3:<VAL>)Theme (x1)Referent)	0
Work	Thing	award	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2: past +WIN_00 (x1)Theme (x3:<VAL>)Referent)	0
Work	Thing	basedOn	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+CREATE_00 (x3)Theme (x1)Referent (f1:(e3: +THINK_00 (x3)Theme (x4:<VAL>)Referent))Reason)	1
Work	Thing	completionDate	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+FINISH_00(x3)Theme (x1)Referent (f1: <VAL> +DATE_00)Time)	1

Work	Thing	genre	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2: +BE_01 (x1)Theme (x3:<VAL>)Attribute)	0
Work	Thing	language	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2: +CREATE_00 (x3)Theme (x1)Referent (f1: <VAL> +LANGUAGE_00)Instrument)	0
Work	Thing	previousWork	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2: past +CREATE_00 (x3)Theme (x1)Referent) +(e3: rpast +CREATE_00 (x3)Theme (x4:<VAL>)Referent)	1
Work	Thing	publishDate	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2: \$PUBLISH_00 (x3)Theme (x1)Referent (f1:<VAL> +DATE_00)Time)	1
Work	Thing	publisher	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2: \$PUBLISH_00 (x3)Theme (x1)Referent (f1:<VAL>)Company)	1
Work	Thing	releaseDate	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+SHOW_00 (x3)Theme (x1)Referent (f1:(e3: ing +BE_01 (x1)Theme (x4:+PUBLIC_00)Attribute))Purpose (f2:<VAL> +DATE_00)Time)	1
Work	Thing	runtime	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+BE_01 (x1)Theme (x3:+DURATION_00)Attribute (f1:<VAL> +SECOND_00)Quantity)	0
Work	Thing	subsequentWork	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) +(e2:+CREATE_00 (x3)Theme (x4:<VAL>)Referent (f1:(e3: past +CREATE_00 (x3)Theme (x1)Referent))Condition)	1

Work	Thing	type	+(e1:+BE_00 (x1: <C>)Theme (x2: +ARTIFICIAL_OBJECT_00 +INFORMATION_OBJECT_00)Referent) *(e2:+BE_01 (x1)Theme (x3:<VAL>)Attribute)	0
------	-------	------	---	---

Leyenda

Domain: clase/subclase de DBpedia a la que pertenece la propiedad

Superclass: hiperónimo inmediatamente superior a la clase/subclase indicada en **Domain**

Property: propiedad

COREL: Regla elaborada en lenguaje COREL

a/n: alternative name

n/r: no rule

Type: tipo de bio-estructura **0** = retrato **1** = historia

APÉNDICE II

Estadísticas totales de la población del Onomasticón

Resumen

Total de propiedades de DBpedia en las que se ha trabajado:	231
Total de propiedades clasificadas como n/r :	90
Total de propiedades clasificadas como a/n :	13
Resto de propiedades para las que se ha elaborado una regla en lenguaje COREL:	128

Recuento total de propiedades COREL: Tipos semánticos

ATRIBUTIVAS SIMPLES	19
ATRIBUTIVAS ABSOLUTAS	3
PREDICATIVAS RELACIONALES	61
PREDICATIVAS INTRÍNSECAS	45
Total	128

Número de apariciones de cada estructura formal en cada uno de los tipos semánticos

		TIPOS SEMÁNTICOS				TOTAL
		ATRIB SIMPLES	ATRIB ABSOL	PRED REL	PRED INT	
ESTRUCTURAS FORMALES	CONTEXT	19	3	61	45	128
	ATTRIBUTE	11	3	6	3	23
	EV+SAT	2	0	15	25	42
	EVENT ROLE	5	3	39	20	67
	TANDEM- CROSS REF	0	0	12	7	19
	EMBED CONDITION	0	0	2	1	3
	RANGE-IN-SAT	0	0	4	5	9
	GENITIVE	1	0	5	1	7

Tabla de aparición porcentual de cada estructura formal, calculada sobre el total de estructuras formales que aparecen en cada tipo semántico

		TIPOS SEMÁNTICOS			
		ATRIB SIMPLES	ATRIB ABSOL	PRED REL	PRED INT
ESTRUCTURAS FORMALES	CONTEXT	100,00%	100,00%	100,00%	100,00%
	ATTRIBUTE	8,59%	2,34%	4,69%	2,34%
	EV+SAT	1,56%	0,00%	11,72%	19,53%
	EVENT ROLE	3,91%	2,34%	30,47%	15,63%
	TANDEM- CROSS REF	0,00%	0,00%	9,38%	5,47%
	EMBED CONDITION	0,00%	0,00%	1,56%	0,78%
	RANGE-IN- SAT	0,00%	0,00%	3,13%	3,91%
	GENITIVE	0,78%	0,00%	3,91%	0,78%

